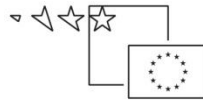




REPUBLIKA SLOVENIJA
MINISTRSTVO ZA ŠOLSTVO IN ŠPORT



Naložba v vašo prihodnost
OPERACIJO DELNO FINANCIRA EVROPSKA UNIJA
Evropski socialni sklad

VZDRŽEVANJE SISTEMSKE PROGRAMSKE OPREME

ROMAN ČUK
JANKO HAREJ

Višješolski strokovni program: Informatika
Učbenik: Vzdrževanje sistemske programske opreme
Gradivo za 2. letnik

Avtorja:

Roman Čuk, univ. dipl. inž. rač. in inf.
Janko Harej, univ. dipl. inž. rač. in inf.
TEHNIŠKI ŠOLSKI CENTER NOVA GORICA
Višja strokovna šola



Strokovna recenzentka:
Barbara Pušnar, univ. dipl. inž. mat.

Lektorica:
Bojana Modrijančič Reščič, prof. slov. j. in dipl. etn. ter kult. antrop.

CIP - Kataložni zapis o publikaciji
Narodna in univerzitetna knjižnica, Ljubljana

004.416(075.8)(0.034.2)

ČUK, Roman

Vzdrževanje sistemske programske opreme [Elektronski vir] :
gradivo za 2. letnik / Roman Čuk, Janko Harej. - El. knjiga. -
Ljubljana : Zavod IRC, 2011. - (Višješolski strokovni program
Informatika / Zavod IRC)

Način dostopa (URL): http://www.impletum.zavod-irc.si/docs/Skriti_dokument/Vzdrzevanje_sistemske_programske_opreme-Cuk_Harej.pdf. -
Projekt Impletum

ISBN 978-961-6857-19-2

1. Harej, Janko
258172928

Izdajatelj: Konzorcij višjih strokovnih šol za izvedbo projekta IMPLETUM
Založnik: Zavod IRC, Ljubljana.
Ljubljana, 2011

Strokovni svet RS za poklicno in strokovno izobraževanje je na svoji 132. seji dne 23.9.2011 na podlagi 26. člena Zakona o organizaciji in financiranju vzgoje in izobraževanja (Ur. l. RS, št. 16/07-ZOFVI-UPB5, 36/08 in 58/09) sprejel sklep št.01301-5/2011/11-2 o potrditvi tega učbenika za uporabo v višješolskem izobraževanju.

© Avtorske pravice ima Ministrstvo za šolstvo in šport Republike Slovenije.

Gradivo je sofinancirano iz sredstev projekta Impletum 'Uvajanje novih izobraževalnih programov na področju višjega strokovnega izobraževanja v obdobju 2008-11'.

Projekt oz. operacijo delno financira Evropska unija iz Evropskega socialnega sklada ter Ministrstvo RS za šolstvo in šport. Operacija se izvaja v okviru Operativnega programa razvoja človeških virov za obdobje 2007-2013, razvojne prioritete 'Razvoj človeških virov in vseživljenjskega učenja' in prednostne usmeritve 'Izboljšanje kakovosti in učinkovitosti sistemov izobraževanja in usposabljanja'.

Vsebina tega dokumenta v nobenem primeru ne odraža mnenja Evropske unije. Odgovornost za vsebino dokumenta nosi avtor.

KAZALO

PREDGOVOR	3
1 UVOD V SISTEMSKO PROGRAMSKO OPREMO	5
1.1 ZGODOVINA SISTEMSKE PROGRAMSKE OPREME	5
1.2 KLASIFIKACIJA SISTEMSKE PROGRAMSKE OPREME	6
1.3 DISTRIBUCIJSKI IN LICENČNI VIDIKI PROGRAMSKE OPREME	12
2 SISTEMSKO SKRBNISTVO	19
2.1 UPRAVLJANJE S ČASOM	20
2.2 NALOGE IN POSTOPKI SISTEMSKEGA SKRBNIKA	22
2.2.1 Podpora uporabnikom	22
2.2.2 Vzdrževanje strojne in komunikacijske opreme	23
2.2.3 Vzdrževanje programske opreme	23
2.2.4 Vzdrževanje storitev	23
2.2.5 Vzdrževanje varnosti	24
2.2.6 Dokumentiranje aktivnosti in vzpostavitev/izvajanje/vzdrževanje ISMS dokumentacije	24
2.2.7 Dela, ki ne spadajo med naloge systemskega skrbnika	25
2.3 IZOBRAŽEVANJE SISTEMSKIH SKRBNIKOV	25
2.4 ITIL (Information Technology Infrastructure Library)	27
3 UPRAVLJANJE PROGRAMSKE OPREME	30
3.1 NAMEŠČANJE IN POSODABLJANJE PROGRAMSKE OPREME	30
3.2 AVTOMATIZACIJA UPRAVLJANJA S PROGRAMSKO OPREMO	32
3.3 SKRIPTNI JEZIKI	33
3.3.1 Tekstovni grafični vmesniki	34
4 VARNE SPLETNE STORITVE	38
4.1 KAKO DO VARNOSTI NA SPLETU	38
4.2 SISTEMSKA PROGRAMSKA OPREMA IN VARNOST	39
4.3 VZPOSTAVLJANJE VARNIH STORITEV	39
4.4 SPREMLJANJE IN NADZOR STORITEV	44
4.5 REŠEVANJE VARNOSTNIH TEŽAV	45
5 VIRTUALIZACIJA	48
5.1 KAJ JE VIRTUALIZACIJA	48
5.2 VIRTUALIZACIJA STROJNE OPREME	50
5.2.1 Oblike virtualizacij strojne opreme	52
5.2.2 Virtualizacija diskovnih polj	54
5.3 VIRTUALIZACIJA NA STRANI DELOVNIH POSTAJ	54
5.4 VIRTUALIZACIJA IN RAČUNALNIŠTVO V OBLAKU	57
5.5 OBLIKE STORITEV	58
5.6 KORAKI PRI UVAJANJU VIRTUALIZACIJE	59
6 UPRAVLJANJE Z VSEBINAMI	62
6.1 UPRAVLJANJE Z VSEBINAMI	62
6.2 VLOGE V ŽIVLJENJSKEM CIKLUSU CMS-JA	64
6.3 SISTEMI ZA UPRAVLJANJE SPLETNIH VSEBIN	65
6.3.1 Uporaba	65
6.3.2 Prenova spletnega mesta podjetja	66
6.4 UPRAVLJANJE Z DOKUMENTI	69
6.5 SISTEMI ZA UPRAVLJANJE ODNOSOV S STRANKAMI	71
6.6 UPRAVLJANJE SISTEMOV ZA UPRAVLJANJE Z VSEBINAMI	72
6.6.1 Nameščanje	73
6.6.2 Vzdrževanje	74
7 LITERATURA	76

PREDGOVOR

Delovno mesto sistemskega skrbnika je zelo lepo in zanimivo ter nikoli dolgočasno. Dnevno omogoča različna stike z uporabniki različnih poklicev in izobrazb znotraj organizacije, kar pozitivno vpliva na osebnostno in strokovno rast. Pri realizaciji kratkoročnih in načrtovanju srednjeročnih ciljev organizacije je nujno tesno sodelovanje vseh vpletenih služb, kar pa omogoča zelo odprt tok informacij. Če je taka oseba tudi odprte narave in pripravljena razvijati svoj potencial, lahko pride do mnogih sinergijskih učinkov ter s tem do uresničitve fraze, da lahko informatika igra ključno vlogo pri razvoju organizacije.

Po drugi strani pa je delo lahko tudi zelo naporno, saj je potrebno pokrivati zelo veliko področij strojne in predvsem programske opreme, ki se v produkcijskih okoljih radi obnašata nekoliko drugače kot v testnih ali laboratorijskih okoljih. Mnoga področja (odvisno od organizacije in stanja v informatiki) moramo podrobno poznati, imeti nekakšen občutek, kako sistem diha in se odziva. Povrh vsega pa se časovna komponenta znanja na tem področju zelo hitro pomika naprej. To sistemskega skrbnika sili v nenehno izobraževanje in iskanje novih rešitev ter pristopov.

Mnenje avtorjev je, da je v takšnem obsegu ur nemogoče podati tako specifična znanja širokemu krogu bralcev. Na takih delovnih mestih je po navadi obdobje uvajanja eno koledarsko leto in še močno je odvisno od števila zaposlenih v informatiki. Zato učbenik poleg svoje širokosti gradi na samostojnem delu študenta, saj bo njegova iznajdljivost tista, ki mu bo pri reševanju težav najbolj koristila. Pri tem pa mu bodo v pomoč predvsem namigi, kako naj na svetovnem spletu (*Internet*) poišče določene rešitve ali namige. Svetovni splet hrani ogromno znanja, vendar je na žalost to znanje izredno slabo organizirano in pri iskanju informacij tudi skoraj vsevedni Google težko ponudi pametne odgovore, če iskalni nizi niso najboljši.

Avtorja predvidevata, da bo preplet večletnih izkušenj z vodenjem informacijskih tehnologij v velikem slovenskem podjetju in večletnih izkušenj s poučevanjem na srednjih in višjih šolah ponudil študentu ustrezen način in obseg snovi v tem učbeniku, da bo lahko z veseljem predelal podano snov in se poglobil v nakazane izzive. Po končanem študiju pa bo uspešno opravljal svoj poklic.

Napotki za uporabo učbenika

Učbenik je sestavljen iz 6 poglavij, v uvodnem je podana klasifikacija sistemske programske opreme. Predstavljena klasifikacija je nekoliko razširjena v primerjavi z mnogimi ostalimi avtorji. Namen te razširitve je dovolj širok pogled na to, kaj vse lahko zajema delo sistemskega skrbnika. Tako ostalih 5 poglavij govori o posameznih področjih, ki jih bo študent lahko srečeval na svoji poklicni poti. Namen števila in širine poglavij je predvsem ta, da študent zazna, da se področje sistemske programske opreme izredno močno prepleta z vsaj tremi ostalimi predmeti na višješolskem študiju. To so operacijski sistemi, varnost in zaščita ter komunikacije. Želja je, da študent vedno razmišlja o sistemskega skrbništva kot prepletu več področij in ne išče samo parcialnih rešitev. Pomoč pri tem je v vsakem poglavju na enem ali več mestih zapis realnih dilem ali problemov, ki se pojavljajo v organizacijah. Ti zapisi imajo namen študenta pritegniti tako, da povezujejo podana znanja s produkcijskim okoljem.

Za lažje sledenje vsebini so pripravljene tudi nekateri poudarki (<http://findicons.com/>, 15. 5. 2011):

Motivacijski element

Tu dobi študent namig ali razlog, zakaj je smiselno poglavju/področju, ki sledi, posvetiti pozornost.



Pomembna informacija

Na tem mestu dajeta avtorja poseben poudarek. Očitno je ta vsebina pomembnejša ali kar ključna za posamezno področje, zato jo je vredno natančno preštudirati.



Uporaba spleta

Znotraj tega bloka se nahajajo namigi, kako začeti z iskanjem na svetovnem spletu in namigi v obliki vprašanj/nalog/izvlečkov/opisov, kaj se tam nahaja. Cilj je, kako se naučiti ločevati zrna od plevel v ogromni množici rezultatov, ki jih ponuja Google.



Študij primera

Povezovanje teorije s prakso je eden od ključnih elementov pri širšem razvoju študenta, saj omogoča, da teoretična znanja, pridobljena s študijem, sooči z realnimi primeri. Izkušnja, ki jo pri tem dobi, je neprecenljive vrednosti pri kasnejšem uspešnem vključevanju v poklic sistemskega skrbnika.



Raziskovanje

Študent bo uspešen, če bo samostojno raziskoval. Tako mu ta blok ponuja temo, ki jo je vredno obdelati.



Učni cilji, ki jih učbenik zasleduje:

- razumevanje pomena in vloge sistemske programske opreme
- vključevanje sistemske programske opreme na različna področja informatike
- podajanje uporabnega in praktičnega znanja o sistemski programski opremi
- razvijanje odnosa do pravočasnega spremljanja novosti na področju vzdrževanja sistemske programske opreme
- razvijanje odgovornosti za načrtno in organizirano delovanje
- razvoj komunikacijskih sposobnosti in spretnosti predstavljanja svojih rešitev

Kaj gradivo ni:

- vseobsegajoči vir;
- edini vir, potreben za uspešen zaključek študija;
- vodič za uporabo določenega operacijskega sistema ali izbrane sistemske programske opreme.

Kaj gradivo hoče biti:

- študijsko gradivo za študente višješolskega programa Informatika,
- prvi usmerjevalec na poteh sistemske programske opreme,
- vodič skozi koncepte upravljanja sistemske programske opreme,
- podlaga za e-učbenik.

Na naslovu <http://pomoc.tsc.si/wiki/VSO> lahko najdete in prispevate dodatna razmišljanja, povezave, dopolnitve, rešitve ipd.

1 UVOD V SISTEMSKO PROGRAMSKO OPREMO

Cilji poglavja

Uvodno poglavje je mišljeno kot pregled celotnega področja sistemske programske opreme – v nadaljevanju tudi SPO. Tu sta zajeta zgodovinski pogled in delitev na področja. Tako nakažemo, kaj so naloge sistemskega skrbnika, na katerih področjih se lahko specializira in v povezavi z licenčnimi vidiki (sistemske) programske opreme tudi na kaj mora biti pri svojem delu pozoren.

Za učinkovito spremljanje vsebin ni potrebnih posebnih znanj. Zadostujeta osnovno znanje uporabe računalnika ter veščina iskanja informacij na svetovnem spletu.



Iz prakse

Bralec lahko v tem poglavju sooči svoja dosedanja praktična znanja iz posameznih področij in jih primerja z zelo obsežnim naborom orodij, programi in okolji, ki jih s tem namenom podajamo v podpoglavju o klasifikaciji. Poudariti velja, da je zelo malo računalniških strokovnjakov, ki bi obvladovali vsa našeta orodja. Ta učbenik tudi ni mišljen, da bi bralec spoznal vsa ta orodja, ampak ga vodi, da dobi širok vpogled v to, kaj je na posameznem področju prisotno ali pomembno. Pridobljena znanja bodo potem bralcu v veliko pomoč, ko se bo v realnih situacijah srečeval s posameznimi orodji, programi ali okolji. Lahko bo lažje komuniciral s sodelavci, razumel njihove poteze in morda dobil celo kakšno idejo katero orodje uporabiti za reševanje problemov s katerimi se bo srečeval.



Kot zanimivost kaj je to sistemska programska oprema si oglejmo zelo znan primer, ko je eden od velikih proizvajalcev programske opreme hotel preko lažnega prikazovanja pomembnosti integracije spletnega brskalnika z operacijskim sistemom vsiliti monopol nad področjem spletnih brskalnikov. Gre za primer brskalnika Internet Explorer in operacijskega sistema Windows 98. Takrat je 12 ameriških zveznih držav tožilo Microsoft, češ da onemogoča konkurenco in s tem izrablja svoj monopolni položaj na tržišču. A je Microsoft trdil, da je integracija Internet Explorerja v operacijski sistem Windows 98 nujna in da brez tega operacijski sistem ne more delovati. Zaradi tega je bilo nameščanje konkurenčnih spletnih raziskovalcev prepovedano oziroma močno oteženo. Sodba, ki je sledila, je določila, da spletni raziskovalci niso del sistemske programske opreme in zato ne smejo biti sestavni del operacijskega sistema. To pa pomeni, da mora biti uporabnikom omogočeno, da to orodje izbirajo po lastni presoji. (http://en.wikipedia.org/wiki/History_of_Internet_Explorer, 17. 1. 2011).

1.1 ZGODOVINA SISTEMSKE PROGRAMSKE OPREME

Prvi elektronski digitalni računalniki so bili tako kot njihovi mehanski predhodniki brez kakršne koli programske podpore. Vnos programov in podatkov se je vršil s pomočjo mehanskih stikal (Tanenbaum, 2001). Taki računalniki so bili največkrat namenjeni izključno za eno samo opravilo kot na primer računanje trajektorij izstrelkov na bojnih ladjah.

Z vedno večjim zanimanjem vojaškega sektorja za uporabo računalnikov se je pojavilo povpraševanje po takih napravah tudi na drugih področjih. Vendar se je zaradi zelo visokih vrednosti takih strojev pojavila želja, da bi lahko računalniki opravljali več različnih nalog in ne samo ene. Tako so se pojavili razni načini, kako program in podatke trajno shraniti in jih takrat, ko to želimo, na preprost in hiter način vnesemo v pomnilnik. Pomembno dejstvo pri tem je tudi, da lahko to opravimo brez napak. Tako so se pojavili prvi načini zapisa programov na papirne trakove in kasneje na luknjane kartice in trakove.

Z vse večjim številom programerjev, ki so ustvarjali programe na takratnih računalnikih, pa so se začele pojavljati tudi prve težave oziroma težnje po poenostavitvi njihovega dela. Hitro so ugotovili, da se veliko njihovega dela podvaja. V tistih časih je moral vsak programer svoj program spisati tako, da je popolnoma vse naredil sam, saj je moral program nadzirati celotno delovanje strojne opreme. Največkrat je to pomenilo branje iz pomnilnih medijev (trak ali kartica), na koncu zapis rezultatov zopet na trak ali kartice, vmes pa so bili klasični algoritmi, kot jih tudi mi poznamo danes. Tako so se pojavile prve rutine, ki so si jih programerji med seboj delili in so predstavljale rutine za branje in pisanje iz pomnilnih medijev. Takih rutin se je oprijelo ime IOCS (*Input Output Control System*) in predstavljajo prve zametke sistemske programske opreme, ki bi jim mi danes rekli gonilniki (Moumina, 2001). Kasneje so se takšne rutine začele zbirati na enem mestu in rodile so se knjižnice.

V desetletjih razvoja računalnikov in računalniških sistemov ter zaradi vedno večjih zahtev, ki so jih imeli uporabniki, so proizvajalci razvili določene tehnike, kako iz strojne opreme iztisniti več, kot bi sicer lahko. Tako so se skozi generacije računalnikov pojavljale razne tehnike, kot so opravila (*jobs*), večopravnost (*multiprograming*), večuporabniški sistemi (*multiuser*), interaktivnost (*interactivity*) in mobilnost (*mobility*). Vsaka od teh tehnik je s seboj prinesla veliko dodatnih rutin, ki so jih programerji med seboj delili in na tak način širili sistemske knjižnice. Te so sčasoma postale tako obsežne, da so se proizvajalci strojne opreme odločili vse rutine, ki so sicer upravljale vire računalnika, združiti v enotno okrilje in s tem doseči lažje in boljše upravljanje z različnimi računalniškimi viri. Tako so se rodili prvi operacijski sistemi, ki so se potem pod pritiski uporabnikov in razpoložljive strojne opreme razvijali do danes in se bodo še naprej (<http://www.osdata.com/kind/history.htm#earliestcomputers>, 1. 2. 2011).

Vzporedno in tudi skladno z razvojem operacijskih sistemov in uporabniških zahtev so se pojavljali tudi programski jeziki. Od povsem prvih zametkov zbirnih jezikov (*assembly language*) v 40. letih, preko proceduralnih jezikov (*procedural language*) 50. in 60. letih. Tipični predstavniki teh jezikov so: Fortran, Algol, Cobol, Basic, Pascal. V 70. in 80. letih so se pojavili objektni jeziki (*object language*), kot so: Simula, Smalltalk, C/C++, Ada. S širitvijo računalniškega omrežja (*internet*) v 90. letih in z množično uporabo svetovnega spleta (*Intranet*) pa so se pojavili nekateri novi objektni jeziki, kot sta Java in C#, z njima in tudi z ostalimi objektnimi jeziki pa so se začeli mešati še skriptni jeziki. Tako so nastali: Python, Ruby, Perl, PHP, ASP, JSP (<http://www.computernostalgia.net/articles/HistoryofProgrammingLanguages.htm>, 3. 2. 2011).

Vedno je veljalo, da so programerji opazili določene lepe lastnosti programske kode oziroma njene uporabe. Delčke teh programov ali kar cela orodja, ki so bila prvotno namenjena za končnega uporabnika, so potem integrirali v posamezne knjižnice, operacijske sisteme ali razna sistemska orodja. Taka programska oprema sedaj ni več namenjena končnemu uporabniku, ampak jo lahko le-ta uporablja pri svojih programih, s katerimi dnevno dela. Tako se je pojavila delitev na uporabniško programsko opremo in sistemska programska opremo.

1.2 KLASIFIKACIJA SISTEMSKJE PROGRAMSKJE OPREME

Kot smo opazili v zgodovinskem pregledu, so se večinoma vsi deli sistemske programske opreme najprej pojavili kot del programske opreme, namenjene posameznemu uporabniku ali rešitvi posameznega problema. Uporaba pa je pokazala, da so taki deli kode lahko zelo koristni vsem uporabnikom, ne samo tistim, za katere so bili pisani. Iz tega vidika tudi izhaja osnovna definicija, kaj sistemska programska oprema sploh je.

DEF 1

Vse, kar povprečen uporabnik računalnika uporablja pri svojem vsakodnevem delu, predstavlja uporabniško programsko opremo (*User Software*). Nasprotno od tega je vse ostalo sistemska programska oprema (*System Software*).



Taka definicija je ob uporabi programske opreme zelo preprosta. Vendar lahko bralca pripelje do dvomljivih odločitev, saj je to le eden od možnih vidikov, kako lahko gledamo na programsko opremo.

Primer (IOCS), ki je opisan v zgodovinskem pregledu, nam lepo razkrije, da uporabniški program ni tisti, ki bi hranil navodila, kako naj se določeni podatki preberejo ali zapišejo na zunanji medij, ampak so ta zelo specifična navodila zapisana v posebnih knjižnicah, ki skrbijo za vhodno/izhodne naprave. Torej uporabnik ni tisti, ki neposredno uporablja vhodno/izhodne knjižnice, pač pa njegov program. Tako lahko vidimo, da so sistemske knjižnice in na splošno sistemska programska oprema samo uslužnostni programi/moduli/strukture/servisi, ki aplikacijskim programom nudijo določene storitve. Sistemsko programsko opremo imamo v računalniku z namenom, da uporabniška programska oprema na lažji in predvsem učinkovitejši način komunicira s strojno opremo računalnika ali s sosednjimi računalniki/napravami.

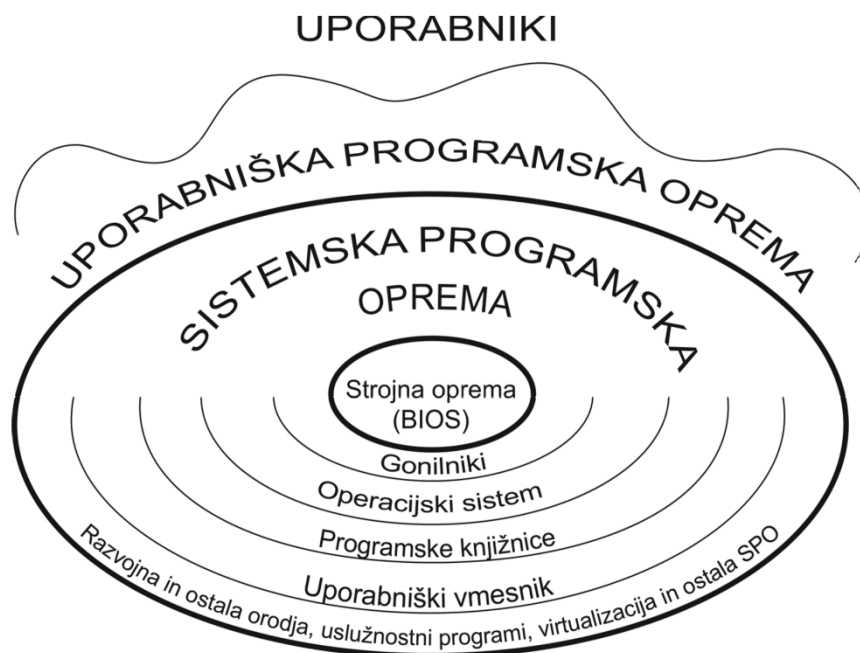
DEF 2

Sistemska programska oprema je tista, ki ostali programski opremi nudi kakršenkoli servis oziroma pomoč.



Iz te definicije lahko hitro opazimo, da programska oprema, ki nudi servis samo uporabniku, spada v skupino uporabniške programske opreme. Poudarek je na besedi »samo«, saj lahko imajo (neposredno ali posredno preko sistemskega skrbnika) korist od servisa tako ostala programska oprema kot tudi uporabnik. Vendar beseda »samo« pomeni izključno samo uporabnik.

Na spodnji sliki lahko vidimo, da uporabnik komunicira izključno z aplikacijsko plastjo, aplikacijska s sistemske in le-ta s strojno opremo. Če pa gremo v obratni smeri, nam strojna oprema zagotavlja storitve sistemske plasti, sistemska plast aplikacijski in slednja zagotavlja storitve uporabniku. Opazimo lahko tudi, da je celo znotraj SPO prisotnih več nivojev različne programske opreme. Bližje kot so nivoji strojni opremi bolj so nujno potrebni za delovanje celotnega računalniškega sistema. Rečemo lahko tudi, so njihove storitve na voljo vedno širšemu naboru programske opreme. Bolj kot se programska oprema oddaljuje od središča proti uporabniku manj programske opreme je odvisno od nje.



Slika 1: Umeščanje programske opreme med uporabnika in strojno opremo

Ker pa imamo lahko težave s tem, kdo je povprečen uporabnik računalnika in kakšne so njegove vsakodnevne naloge, je smiselno definicijo še nekoliko razširiti.

DEF 3

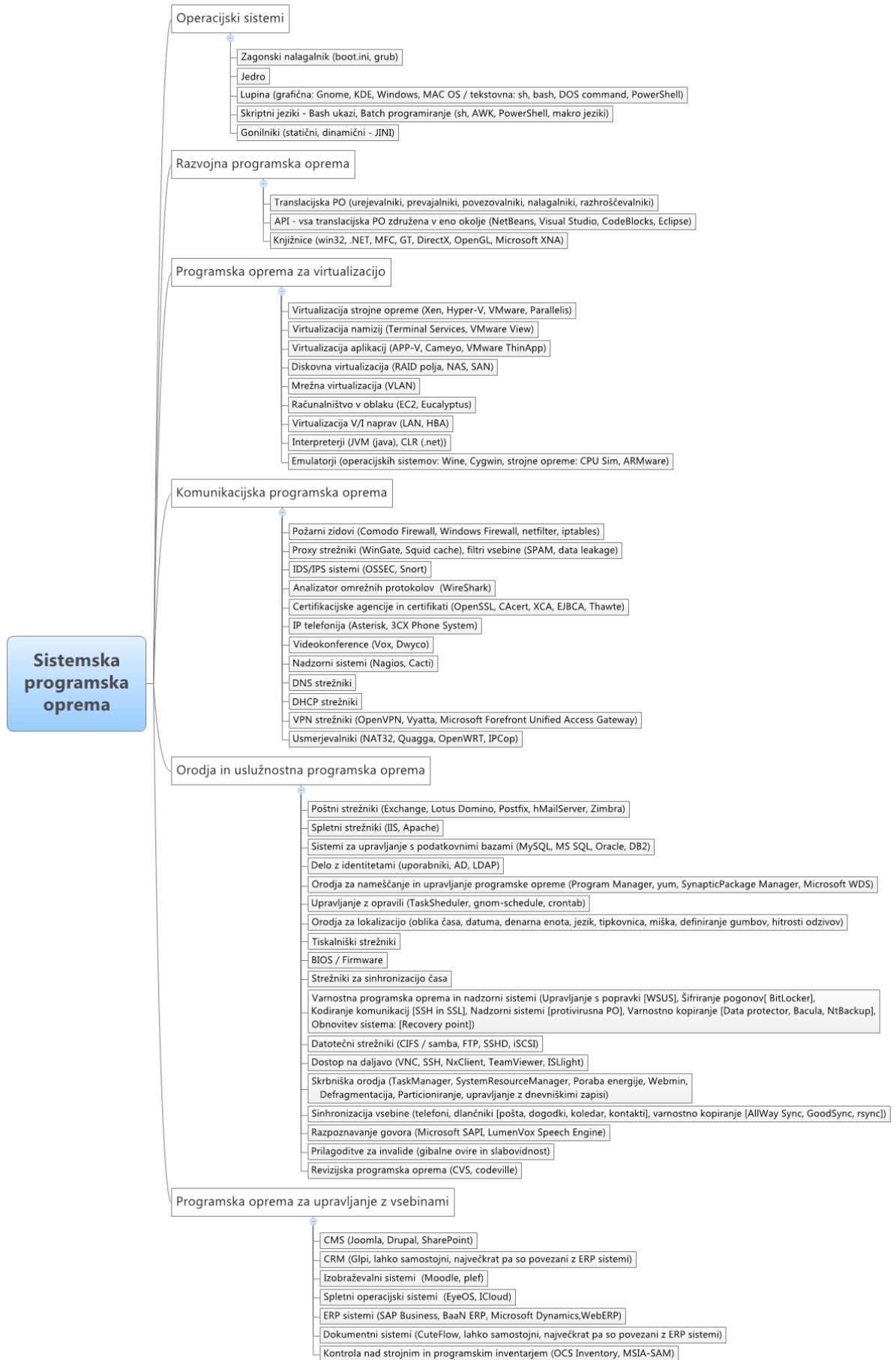
Sistemska programska oprema nudi okolje za izvajanje aplikacij, razvoj le-teh in zagotavlja izvajanje storitev.



Taka definicija je smiselna tudi zaradi tega, ker je na mnogih področjih meja med sistemsko in uporabniško programsko opremo zelo tanka in se v desetletjih ter z načinom ali razlogom uporabe računalnikov spreminja.

Po drugi strani pa nam taka definicija tudi jasno pokaže, s katerimi dejanskimi programskimi produkti se dnevno ukvarja sistemski skrbnik (*System Administrator*) ali razvijalec programske opreme (*Programmer*) in s katerimi se ukvarja povprečen uporabnik (*User*). Kot zanimivost pa povejmo še to, da tudi uporabniki ali skrbniki v odvisnosti od njihove funkcije v organizaciji različno gledajo na to področje. Kot primer pogledjmo samo poljubno manjšo organizacijo, kjer je uporabnik programske opreme istočasno lahko tudi skrbnik sistema in še razvijalec.

V nadaljevanju si na spodnji sliki oglejmo, kakšna je podrobnejša klasifikacija sistemske programske opreme, na katera ključna področja je razdeljena in kateri so značilni predstavniki v teh skupinah.



Slika 2: Klasifikacija sistemske programske opreme

V skupini **operacijski sistemi** (*Operating System*) so združene najosnovnejše metode/operacije za delo z računalnikom oziroma za zagotovitev okolja, v katerem se lahko katerakoli aplikacija sploh izvaja. Tu se nahajajo zagonski nalagalniki, ki poskrbijo za to, da se operacijski sistemi sploh lahko začnejo nalagati. Potem je tu jedro kot osrednji del operacijskega sistema, ki skrbi za dodeljevanje virov. Z vidika uporabnika je najpomembnejša lupina, ki jo ima posamezni sistem. Tu gre največkrat za grafično prijazno okolje, v katerem se uporabnik znajde, in največkrat poistoveti vrsto operacijskega sistema prav z grafičnim okoljem, kjer dela, oziroma s tekstualnim okoljem (lupino), če nima grafične podpore. Tej uporabniški izkušnji potem sledijo razna orodja za pomoč pri delu z lupinami, kot so makro jeziki (*macro language*) v grafičnih okoljih in skriptni jeziki (*scripting language*) v tekstovnih okoljih. Na koncu pa je potrebno omeniti še gonilnike (*driver*), ki skrbijo za komunikacijo med strojno opremo in jedrom operacijskega sistema.

V skupini **razvojna programska oprema** (*Software development*) so orodja in okolja za razvoj kakršnihkoli programov. Podan je pregled, katere korake mora izvorni program preiti, preden se lahko brez napak izvaja v pomnilniku. To so: pisanje programa, prevajanje, povezovanje, nalaganje in razhroščevanje. Na koncu so predstavljena tudi integrirana okolja za razvoj programov, v katerih so vsi ti koraki pravzaprav avtomatizirani in si sledijo eden za drugim.

V skupini **programska oprema za virtualizacijo** (*Virtualization software*) so orodja, ki nam omogočajo virtualizacijo izbranega področja. To je izvedeno na tak način, da se med strojno opremo ali delom programske opreme in subjektom virtualizacije namesti nekakšen virtualizacijski sloj (*Hypervisor*), ki je lahko programski ali strojni in nam omogoča, da s tem preslepimo tisto programsko opremo, ki jo želimo virtualizirati. Dejansko subjekt virtualizacije niti ne opazi, da se ne izvaja na pričakovanem temelju, ampak se v resnici izvaja v navideznem okolju. Področja, ki jih lahko virtualiziramo, so:

- Na strani **strojne opreme**, kjer lahko virtualiziramo: operacijske sisteme, diskovni prostor (RAID, SAN, NAS), računalniško mrežo (VLAN), Vhodno/Izhodne naprave, izvajalno okolje.
- Na strani **uporabnika**, kjer lahko virtualiziramo: namizje uporabnika in aplikacije, ki se pri njem izvajajo.

V skupini **komunikacijska programska oprema** (*Communication software*) so orodja za zagotavljanje delovanja interneta, lokalnih omrežij in povezovanja le-teh. Nadalje spadajo sem orodja za zagotavljanje raznih komunikacijskih servisov (telefonija, videokonference, FAX), nadzornih sistemov in zagotavljanja varnosti na prenosnem mediju ter točkah prehoda. Poudariti velja, da je tu poudarek izključno na programski opremi, vendar za podrobno razumevanje tega področja potrebujemo tudi znanja iz protokolov, ki komunikacije sestavljajo. Protokoli, kot so IPsec, SSL/TLS, DNS, DHCP, HTTP, IMAP, POP, SMTP, so podrobneje obravnavani pri predmetu Komunikacije, zato se z vidika vzdrževanja sistemske programske opreme v ta področja ne spuščamo.

V skupini **orodja in uslužnostna programska oprema** (*Administrative tools, Services*) so orodja, s katerimi si skrbnik sistema pomaga zagotavljati ustrezno delovanje celotnega računalniškega sistema – od strojne do programske opreme. To pomeni od raznih nastavitve jezika, gumbov na tipkovnici in miški, hitrega in ustreznega delovanja diskovnega sistema pa vse tja do upravljanja z (še ne) nameščeno programsko opremo in zagotavljanjem varnosti na računalniškem sistemu s stališča popravkov kot tudi zaščite shranjenih podatkov. V drugem delu skupine pa se skrivajo servisi, ki uporabnikom ali sistemu samemu zagotavljajo določene storitve. Primer za to so poštni, spletni, podatkovni, tiskalniški, datotečni strežniki ter ostali

strežniki, ki nam zagotavljajo nadzor nad nekim področjem ali pa nam zagotavljajo neke prilagoditve, s pomočjo katerih lahko uporabniki dostopajo do različnih računalniških virov.

V skupini **programska oprema za upravljanje z vsebinami** (*Content Management Software*) so orodja/okolja, ki nam zagotavljajo razne storitve. Te so upravljanje z vsebinami na spletnih portalih, upravljanje z odnosi do naših strank, dokumentni sistemi, izobraževalni sistemi, kontrola nad inventarjem ter sistemi za vodenje podjetij. Vsa ta okolja delujejo na tri nivojski arhitekturi, kar pomeni, da imamo na eni strani podatkovno bazo, ki hrani za nas pomembne podatke, na drugi strani imamo aplikacijo, ki nam zagotavlja pravilne vnose in prikaze te vsebine ter obdelavo podatkov in na tretji strani imamo uporabnika, ki dostopa do aplikacije.

Velika večina omenjene sistemske programske opreme je takšna, da lahko brez težav trdimo, da spada v določeno skupino. Določena oprema pa je takšna, da bi jo lahko uvrstili v dve področji ali več podpodročij. Primer za to bi lahko bilo upravljanje z identitetami. Identitete so dejansko uporabniki, ki jih obravnavamo kot računalniški vir in zato spadajo pod operacijske sisteme. Vendar če imamo večjo organizacijo, kjer ima dostop do računalniškega sistema mnogo uporabnikov (več sto ali celo tisoč), moramo na identitete gledati povsem drugače. V takih okoljih z lokalnimi uporabniki ne moremo zagotavljati ustrezne varnosti in nadzora nad dokumenti/aplikacijami ali pa vsaj ekonomsko to ni upravičeno. V takih primerih potem govorimo o imenikih oziroma imamo servis, ki nam zagotavlja avtentikacijo in avtorizacijo uporabnikov in lahko tudi ostalih virov. Drugi primer pa bi lahko poiskali iz področja komunikacij in varnosti kot na primer varni komunikacijski kanali.

Razprava med študenti

Študentje naj se med seboj pogovorijo, ali je morda kateri od programov/orodij /okolij, navedenih v klasifikaciji, razvrščen v napačno skupino. Želja je, da študentje ugotovijo, da lahko določena orodja spadajo v dve ali več skupin. Podrobneje naj se pogovorijo o teh dilemah. Ne vztrajati pri trditvi pravilno/napačno, poudarek je na argumentiranju.



Zaradi lažjega osvajanja te klasifikacije je smiselno predstaviti tudi njen antipod (http://en.wikipedia.org/wiki/Application_software, 5. 2. 2011). Torej kaj sistemska programska oprema ni! Če samo naštejemo: oblikovalniki besedil (Word, Writer), elektronske preglednice (Excel, Calc), predstavitvena orodja (PowerPoint, Presenter), spletni brskalnik (Internet Explorer, Firefox), programi za prikaz/obdelavo multimedijskih vsebin (Media Player, VLC, Audacity, Adobe Photoshop, Pinnacle Studio), simulacijska orodja (Fluent, Magma, Flux), CAD/CAM sistemi (Acad, ProEngineer, Mastercam), odjemalci spletne pošte (Outlook, Thunderbird), sistemi za takojšnje sporočanje (Windows Live Messenger, ICQ, Skype, Tencent QQ), igre (Super Mario, Tetris, Angry birds).

Izpostaviti je potrebno še to, da ni poudarek na poznavanju klasifikacije na pamet. Tudi različni avtorji podajajo klasifikacijo na različne načine¹. Naša želja je, da se bralec zaveda, kakšno delo opravlja sistemski skrbnik, na kaj vse mora biti pozoren pri svojem delu in pravzaprav tudi na kakšen način oziroma za katera področja se sistemski skrbnik specializira. Seveda slednje velja, če gre za večje organizacije. V manjših organizacijah je praviloma zaposlen samo en sistemski skrbnik in še ta ima mnogokrat tudi druge zadolžitve. Tako ta skrbnik vedno opravlja veliko stvari naenkrat in v nobeno se ne more poglobiti tako, kot bi si sam želel.

¹ (http://en.wikipedia.org/wiki/Category:System_software, 5. 2. 2011)

(http://en.wikipedia.org/wiki/System_software, 5. 2. 2011)

(http://www.swbusiness.fi/attachments/software_classification_explanation.pdf, 5. 2. 2011)

Razprava med študenti

Razmislite o tem, kako je program v drugem letniku višješolskega študija (modul Sistemsko administriranje in modul Upravljanje podatkovnih omrežij) razdeljen po predmetih. Ali vidite kakšno povezavo s klasifikacijo SPO?



1.3 DISTRIBUCIJSKI IN LICENČNI VIDIKI PROGRAMSKE OPREME

Vsaka programska oprema, sistemska ali uporabniška, mora biti legalna. Če se v domačih okoljih na nek način redno ne kontrolira legalnosti programske opreme, je v podjetjih legalnost pravilo (Trampuš, 2004). Zaradi tega in tudi zaradi visokih kazni za podjetje in za uradno ter odgovorno osebo je nujno poznati določene pojme v zvezi z licenčnimi pogoji uporabe programske opreme ter možnosti pridobitve, uporabe, spreminjanja in morebitnega posredovanja tretjim osebam.

Sodna praksa

http://www.sodisce.si/znanje/sodna_praksa/upravno_sodisce_rs/2010040815244047/ (7. 3. 2011) je naslov, kjer lahko bralec najde primer sodne prakse na področju nelegalne uporabe lastniške programske opreme. Bodite pozorni na postopek, kako lahko inšpektor pregleda, katera programska oprema je nameščena. Kako se obravnava operacijski sistem? Kakšen je postopek po ugotovljenih nepravilnostih?



Licence se delijo na tri velike skupine (http://en.wikipedia.org/wiki/Software_license, 17. 3. 2011):

- lastniške (zaprta koda),
- proste (odprta koda in zaprta koda),
- javne (odprta koda in zaprta koda).

Pri vseh licencah so pravice in dolžnosti, ki jih ima končni uporabnik bolj ali manj zapletene, vendar dobro opisane. Dodatno vsak avtor določene stvari, kot so pridobitev, uporaba, možnost spreminjanja in posredovanja sprememb uredi nekoliko po svoje. Zaradi tega hitro nastane pri uporabi programske opreme velika zmešnjava, kaj sistemski skrbnik ali domači uporabnik lahko uporablja in kako ter česa ne sme. Dodatno obstaja še delitev na odprto in zaprto kodo. Če avtor omogoči dostop do izvirne kode programa, to pomeni odprta koda. Če pa tega ne stori, pravimo, da je koda zaprta. Dostop do izvirne kode je lahko javen, lahko pa je omogočen samo tistim, ki so pridobili pravico uporabe določenega izdelka.

Razprava med študenti

Izmenjajte mnenja o tem, v katerih primerih je možno izvesti prenos licenc na drugo osebo (pravne in/ali fizične osebe).



V nadaljevanju bomo v tabeli prikazali nekaj najznačilnejših licenc in kako jih lahko uporabljamo². V vsakem primeru je potrebno poudariti, da neglede na uporabljeno licenco leta podeljuje avtorju izključno pravico nad avtorstvom. To se kaže s tem, da je ime avtorja (jev) ali logotip podjetja viden ob zagonu programa ali pa je uporabniku pri uporabi programa omogočeno, da te podatke vidi. Največkrat je to izvedeno pri izbiri »Vizitka« (*About*) v meniju »Pomoč« (*Help*). Teh podatkov ne sme nihče nikoli spreminjati, lahko pa doda svoje ime in nakaže avtorstvo ob njegovih spremembah.

² (http://en.wikipedia.org/wiki/Berkeley_Software_Distribution, 17. 3. 2011)

(http://en.wikipedia.org/wiki/GNU_General_Public_License, 17. 3. 2011)

(<http://creativecommons.org/licenses/>, 17. 3. 2011)

(<http://en.wikipedia.org/wiki/Shareware>, 15. 3. 2011)

(http://en.wikipedia.org/wiki/Criticism_of_Microsoft, 17. 3. 2011)

Skozi »Vizitko« je v mnogih primerih lastniške opreme omogočen tudi dostop do dejanske licence, v kateri je pravno gledano zapisan odnos med lastnikom programske opreme in končnim uporabnikom te opreme. Poudariti je potrebno, da ima ta zapis povsem enako pravno vrednost, kot jo ima pogodba. Končni uporabnik »podpiše« pogodbo s tem, ko sprejme licenčne pogoje.

Zmeda ali red?

Na naslovu <http://www.gnu.org/licenses/license-list.html> (17. 3. 2011) si lahko ogledamo enega od seznamov licenc. Tu so zbrane tako prosto dostopne kot tudi lastniške licence. Če bi jih šteli, bi našli okrog 100 različnih vrst licenc (vključno z verzijami). Če nekoliko razmislimo in sprejmemo možnost, da ima vsak od stolpcev v tabeli samo 5 različnih možnosti, to pomeni, da imamo lahko 625 različnih vrst licenc.

Največ možnosti za vpeljavo svojih rešitev imamo ravno na področju spreminjanja in posredovanja, tako da ni težko videti, da je glavnina »zmede« prav v področju proste programske opreme.



Zanimivo branje

Bralec naj si poišče članek Vladimirja Djurdjiča v reviji Monitor z naslovom Odrpta državna uprava (Djurdjič, 2011). Tu avtor podaja argumentirano kritiko na študijo(e) uvajanja odprte kode v državno upravo. Ravno tako je zanimiv tudi nekoliko bolj poglobljen članek Horuk v odrpto kodo avtorja Daret Hriberška v reviji Monitor PRO (Hriberšek, 2011). Tu se lepo vidi, da se lahko odločitve sprejema tudi na podlagi kakšnih drugih parametrov, kot je samo cena.

Dodatno naj si bralec za kritično oceno uvajanja odprte kode ogleda še sledeče povezave (vse: 15. 3. 2011):

- http://www.mju.gov.si/fileadmin/mju.gov.si/pageuploads/mju_dokumenti/pdf/STUDIJA_ODPRTA_KODA.pdf
- http://www.pcworld.com/article/131533/berlin_says_nein_to_opensource_migration.html
- <http://www.muenchen.de/Rathaus/dir/limux/english/147197/>
- <http://www.muenchen.de/limux>
- <http://www.h-online.com/open/news/item/LiMux-project-management-We-were-naive-958824.html>
- <http://dk.fdv.uni-lj.si/diplomska/pdfs/upelj-boris.pdf>
- <http://slo-tech.com/novice/t456705/0>
- <http://www.h-online.com/open/news/item/German-Foreign-Office-explains-open-source-elimination-1241804.html>



Bodite pozorni na dejstvo, da prehod ni samo namestitev programske opreme, ampak predvsem izredno velik projekt izobraževanja, prilagajanja ter tudi vzdrževanja le-te. Uspeh pa je pogosto odvisen tudi od sprejemanja celotne javnosti. Pomislite samo na to, kdaj ste svojim prijateljem ali sosedom nazadnje popravili računalnik (namestitev razne programske opreme) in zakaj pri tem niste uporabili odprte kode.

Tabela 1 : Prikaz povezave med različnimi vrstami licenc in različnimi pravicami/dolžnostmi zaradi uporabe programske opreme. Siva barva predstavlja proste licence, modra javne licence in rdeča lastniške licence.

Vrsta licence	Pridobitev	Uporaba	Spreminjanje	Posredovanje
Freeware (zaprta koda)	Prosta, na svetovnem spletu. Največkrat je funkcionalnost neokrnjena.	Avtor določi, na katerih področjih dovoljuje uporabo. Največkrat je oprema odprta tako za zasebno kot tudi komercialno uporabo.	Največkrat ni dovoljeno. Tudi redko se zgodi, da imamo na razpolago izvorno kodo.	Da, lahko je celo zaželeno.
GPL (<i>General Public License</i>) (odprta koda)	Prosta, na svetovnem spletu.	Poljubna	Poljubno. Vendar moramo biti zelo pozorni, če kasneje želimo spremembe posredovati naprej. Pri dodajanju novih funkcionalnosti lahko pride do mešanja kode, ki je izdana pod različnimi licencami, ki pa najverjetneje nimajo enakega pogleda, kako in pod kakšnimi pogoji lahko spremembe – torej naše delo – posredujemo naprej. Licence si tu lahko celo nasprotujejo.	Da. Sprememba mora biti izdana pod enakimi pogoji, kot je bila pridobljena. Če izdamo program v binarni obliki, je potrebno omogočiti dostop tudi do izvorne kode ali celotni javnosti ali samo posameznikom, ki našo »spremembo« kupijo. Spremembe, ki smo jih vnesli v obstoječo kodo, lahko tudi tržimo. Največkrat se to doseže z nakupom dovoljenja za uporabo ali pa nakupom podpore.
BSD (<i>Berkeley Software Distribution</i>) (odprta koda)	Prosta, na svetovnem spletu	Poljubna	Poljubno	Da. Vendar ta licenca dovoljuje tudi spremembo pravic. Še več, kodo lahko tudi zaklenemo. Na tak način je izdan tudi Mac OS, ki prvotno izhaja iz FreeBSD.
CC (<i>Creative Commons</i>) (odprta koda ali zaprta koda)	Ni pomembno. Največkrat na svetovnem spletu. Ta licenca se uporablja pretežno pri izdaji multimedijских gradiv (slike, ikone), učnih gradiv (ta učbenik) in izobraževalnih programov. Namen licence je, da se vsebino,	Največkrat je poljubna. Lahko pa avtor ravno tako kot drugje omeji nekatera področja. Na primer: prepovedano za komercialno uporabo, prepovedano spreminjanje.	Največkrat je dovoljena. V primeru izrecne prepovedi spreminjanje ni dovoljeno.	Spremembe (ali lahko tudi brez) je dovoljeno posredovati. Vendar avtor lahko zahteva posredovanje pod enakimi pogoji. Če prepoved ni navedena, se lahko posredovanje izvede tudi pod drugačnimi pogoji – lahko tudi

	izdano pod to licenco, uporablja v čim širšem obsegu. Redkeje se licenco CC uporablja za izdajo programske opreme, saj je ta licenca zelo podobna licenci BSD in/ali GPL, ki pa posamezne pravice podrobneje urejata.			kot zaprta koda.
Shareware (zaprta koda)	Prosta, na svetovnem spletu. Licenca praviloma velja samo določen čas in je namenjena testiranju. Po poteku tega obdobja je potrebno licenco za uporabo kupiti ali program odstraniti iz računalnika. Možna je tudi varianta, da je funkcionalnost programa omejena, posledično pa ni časovne omejitve. Namen je torej testiranje in če smo zadovoljni, kupimo.	Avtor določi, na katerih področjih dovoljuje uporabo. Največkrat je to samo domača uporaba ali za namen izobraževanja. Ob nakupu postane to vprašanje nepomembno.	Ni dovoljeno. Tudi izvorna koda ni javna.	Načeloma da, ker je program v polni in okrnjeni različici enak. Uporabnik samo z vnosom ustrezne kode »aktivira« določene funkcionalnosti. Če so polno in neomejeno verzijo programa dostavili avtorji, posredovanje ni dovoljeno.
EULA (<i>End-user license agreement</i>) (zaprta koda)	Nakup licence pri uradnih prodajalcih ali zastopnikih. Tukaj gre največkrat za fizično obliko CD ali DVD plošče. Lahko pa dobimo opremo že pred nameščeno ob nakupu strojne opreme (OEM). Redko se dogaja, da lahko dobimo program tudi na spletu.	Strogo je določeno, kdo lahko opremo uporablja in kako. Pri prvi možnosti omejevanja gre za omejitve števila uporabnikov, oziroma se jih omeji samo na tiste, ki imajo legalen fizični dostop do računalnika. V drugem primeru pa se omejitev lahko izvede na podlagi strojne opreme – primer je število jeder ali procesorjev. Ob nakupu ni pomemben namen oziroma področje uporabe	Ni dovoljeno. Tudi izvorna koda ni javna. Še več, celo posnemanje oziroma povratni inženiring ni dovoljen.	Ne.

		programske opreme.		
--	--	--------------------	--	--

Nakup licenc je mnogokrat povezan z velikimi finančnimi sredstvi, ki jih moramo zagotoviti v enkratnem znesku in ti zneski so lahko že v primeru srednjega podjetja (300 uporabnikov) hitro večji kot 100.000€. Dodatno je tu težava v tem, da si vsak sistemski skrbnik želi imeti samo eno različico programske opreme. To močno olajša vzdrževanje te opreme in tudi cenovno predstavlja veliko notranjo rezervo v podjetju. Težava pa se pojavi, ker z nakupom licenc ne moremo nadgrajevati programske opreme drugače kot z novim nakupom. Še en vidik, ki podjetja sili v iskanje drugačnih rešitev, je dejstvo, da se podjetje lahko širi. Če hoče imeti vse računalnike in uporabnike licenčno pokrite, je prisiljeno kupovati licence po potrebi, kar pa hitro pripelje do vzdrževanja različnih verzij istega produkta.

Rešitev takih dilem je v najemu programske opreme. Sicer je v vsakem primeru najem dražji od nakupa, vendar tak način v kombinaciji z dovoljenjem uporabe vedno najnovejše različice programske opreme za mnoga podjetja predstavlja zelo mikavno rešitev. Tak način nam poleg enotne opreme in lažjega povečevanja števila uporabnikov in/ali računalnikov prinese tudi povsem drugačen finančni vidik. Najem je namreč lahko izveden v več obrokih, kar podjetja zelo cenijo. Možno pa je izvesti celo leasing najema. V takem primeru za podjetje pomeni najem opreme strošek in ne investicije, kar s finančnega področja v podjetju lahko predstavlja celo ugodnejšo rešitev kot nakup.

Najem ali nakup – kakšen je strošek?

V podjetju, kjer opravljate prakso, povprašajte, ali so že razmišljali o najemu programske opreme. Če so, jih prosite, da vam pokažejo ponudbe, ki so jih takrat pridobili, če pa tega niso nikoli počeli, pa jim predlagajte, da vi naredite to zanje. Težava je namreč v tem, da lahko te podatke pridobite samo od pooblaščenih prodajalcev ali proizvajalcev opreme. Kakršnih koli uporabnih števil na spletu namreč ni mogoče dobiti.

Kakšno časovno obdobje morate obravnavati, da dobite uporabne rezultate? V kakšnem časovnem intervalu postane najeta oprema »dražja« od kupljene? Kakšno vlogo igra pri teh izračunih TCO – celotni stroški lastništva (http://en.wikipedia.org/wiki/Total_cost_of_ownership, 13. 3. 2011)?



Kaj nas čaka v prihodnosti?

Trenutno veljavni licenčni vidiki so primerni za posamezne uporabnike in tudi velika podjetja. Strogo spoštovanje licenčnih vidikov pa se začne na področju računalništva v oblaku kazati kot velika ovira pri izkoriščanju vsega potenciala, ki ga računalništvo v oblaku prinaša. Nujno bo potrebno razmišljati o spremembi licenčnih pravil. Vendar s stališča uporabnika storitev v oblaku bo to problem za ponudnika.

Kakšne so te ovire, lahko bralec začuti ob prebiranju spodnjega članka: <http://devcentral.f5.com/weblogs/macvittie/archive/2009/01/27/cloud-computings-other-achilles-heel-software-licensing.aspx> (28. 4. 2011).



Povzetek

Razumevanje pojma sistemska programska oprema nam omogoča spoznati področja, na katerih je prisotna, in vzroke za njen nastanek ter delitev. Taka področja nam pozneje definirajo naloge, ki jih sistemski skrbnik mora izvajati na posameznem področju. Torej predmet Vzdrževanje sistemske programske opreme usmerja bralca k nalogam, ki jih mora sistemski skrbnik izvajati, s tem pa ga pripravlja na poklic sistemskega skrbnika. Natančnejša delitev nalog/postopkov/področij ki se sistemskega skrbnika dotikajo so opisana v ostalih poglavjih tega učbenika.

Poleg uspešnega opravljanja nalog mora sistemski skrbnik dobro poznati tudi pravno področje, ki pokriva lastništvo programske opreme, ki jo pri svojem delu uporablja oziroma posreduje svojim zaposlenim v podjetju.

Vprašanja in naloge

- 1) Opišite, kako je prišlo do pojavitve sistemske programske opreme. Razmislite kakšen pomen je imela SPO včasih in kakšen je njen pomen danes.
- 2) Zakaj se uporablja sistemska programska oprema?
- 3) Naštejte glavna področja, na katera se deli sistemska programska oprema ter področje na kratko opišite. Poseben poudarek dajte na področja ki se prekrivajo.
- 4) Kaj s pravnega vidika pomeni licenca?
- 5) Na katera velika področja se delijo licence?
- 6) V urejevalniku besedila Word (po želji izberite iz katerega Office paketa) poiščite, pod kakšno licenco je izdan Word (Office) in jo dobro preučite. Licenca ima status pogodbe in vi ste se z njeno vsebino strinjali (tisti, ki imate nameščen Office paket). Ali spoštujete vse omejitve?
- 7) Če bi se odločili za izdelavo večjega programskega izdelka, pod katero vrsto licenc bi izdelek ponudili trgu?
- 8) Kaj je glavni razlog, da se podjetja odločajo za najem programske opreme?

2 SISTEMSKO SKRBNIŠTVO

Cilji poglavja

Glavni cilj poglavja Sistemsko skrbništvo je spoznati, s kakšnimi težavami se sistemski skrbnik (*System Administrator*) srečuje pri svojem vsakodnevnem delu. Večino večšin, ki jih pričakujejo od skrbnika sistema, se lahko bralec nauči v procesu izobraževanja ali sam s pomočjo spleta. To so predvsem tehnične veščine. Določen del večšin pa je takih, ki se jih ne da naučiti, ampak se jih je potrebno priučiti v podjetju. Potrebno je spoznati, da posameznik ali skupina mnogokrat ne zmoreta vsega storiti sama in zato potrebujeta nekakšno pomoč v obliki drugačnega pristopa pri odpravljanju težav in reševanju nalog.

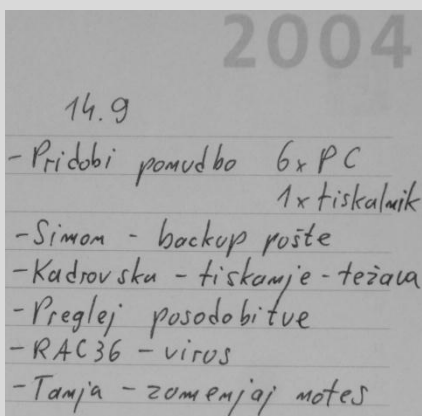


Za učinkovito spremljanje vsebine poglavja ni potrebnih posebnih predznanj, je pa priporočljivo poznati način dela v ukazni vrstici poljubnega operacijskega sistema (*user shell*) in orodja za avtomatizacijo opravil v grafičnih okoljih. Podrobneje o tehničnih rešitvah tu nakazanih problemov govorijo naslednja poglavja tega učbenika.

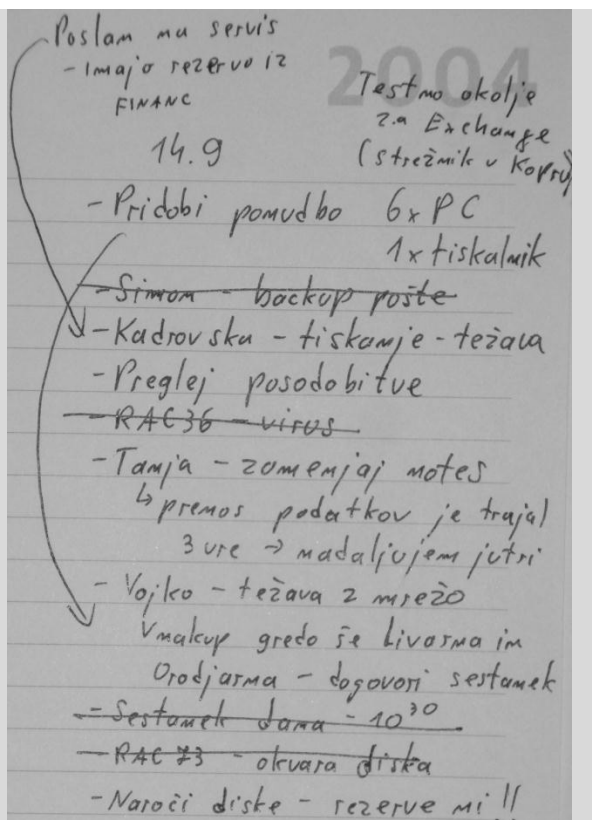
Skrbnik, ki mu uspe najti pravi način, kako v veliki količini dela ohraniti dovolj časa in energije za dodatno izobraževanje, lahko pri svojem delu še dodatno napreduje in tako postaja vedno bolj izkušen. Na tak način sistemski skrbniki pridobivajo pri svojem ugledu, nadrejeni pa jim zaupajo vedno kompleksnejše naloge. Tako skrbnik napreduje po lestvici od začetnika (*Junior System Administrator*) do sistemskega arhitekta (*Senior System Administrator*).

Iz prakse

Seznam opravil ob začetku delovnega dne in seznam ob koncu. Končani sta samo dve opravili, čeprav smo jih načrtovali šest. Dodatne izvedene aktivnosti so povzročile opustitev nekaterih manj pomembnih opravil, ki se prestavijo na naslednji dan.



Slika 3: TODO liste



Kako si pomagati?

Na spletnem naslovu <http://video.google.com/videoplay?docid=7278397109952382318#> (3. 3. 2011) si oglejte video posnetek nastopa Thomasa A. Limoncellia. To je eden od mnogih sistemskih skrbnikov pri Googlu. Bodite pozorni na metode in načine realizacije, ki jih predlaga za lažje in učinkovitejše delo ter se sprašuje, katere so največje težave, s katerimi se srečujejo sistemski skrbniki.



2.1 UPRAVLJANJE S ČASOM

Sistemski skrbniki se dnevno srečujejo z mnogimi izzivi. Nekateri med njimi so načrtovani, velika večina ostalih pa je nenačrtovanih ali vsaj delno nenačrtovanih. Posledica nenačrtovanih aktivnosti je največkrat zamujanje rokov, ki jih skrbnik sicer načrtuje (Limoncelli, 2005). Torej na račun nepričakovanih dogodkov potem trpi projektno delo. Če se takšno stanje dogaja dalj časa, se pri skrbnikih začne pojavljati preobremenjenost in v nadaljevanju lahko to pripelje tudi do nezadovoljstva med uporabniki. Zavedati se je namreč potrebno, da je sistemski skrbnik v organizaciji zadolžen izključno zato, da tam zagotavlja servis vsem ostalim zaposlenim. Ključno vodilo znotraj organizacije je namreč izpolnjevanje njenega poslanstva in informatika je pri tem samo sredstvo za lažje doseganje skupnih ciljev. Torej če nekdo od zaposlenih ne more opravljati svojega dela zato, ker mu računalnik ali kaj v zvezi z njim ne deluje, je za to odgovoren skrbnik.

Če je v ekipi informatikov zaposlenih več skrbnikov, je možno delo prerazporejati in s tem omogočiti posameznikom, da ostanejo osredotočeni na projektno delo. Če pa je skrbnik edini v organizaciji, ki je zadolžen za delovanje celotnega računalniškega sistema, je prepuščen na milost in nemilost uporabnikom in svoji iznajdljivosti. Na srečo obstajajo tehnike, kako si pri tem pomagati in se s tem izogniti, da nas nenačrtovani dogodki ne onesposobijo.

Izziv

Pri informatiku v podjetju se je oglasil vodja proizvodnje in povprašal kdaj bo popravljen tiskalnik na eni od proizvodnih linij. Delavci so napako prijavili že pred tremi dnevi. Razložite mu, da imate zelo veliko dela in da bo njihov tiskalnik šel pogledati jutri, sploh pa nima na zalogi nobenega rezervnega tiskalnika. Vodja proizvodnje pa nadaljuje, da gre pošiljka 20 palet čez 4 ure iz hiše. Paleta morajo biti nujno opremljene s spremno dokumentacijo iz katere je razvidna sledljivost. Dokumentacijo v elektronski obliki sicer imajo, vendar upa da jo ne bodo pomešali med paletami.

Razmislite, kakšen pomen lahko imajo tudi navidez nepomembni dogodki. Kako se bo informatik sedaj urgentno lotil reševanja nastale situacije. Kaj se lahko zgodi, če bo dokumentacija zamešana? Razmislite kaj lahko to pomeni za kupca in kaj za vaše podjetje pomeni reklamacija? Take posledice je možno finančno ovrednotiti. Kaj pa pomeni okrnitev dobrega imena za nadaljnje posle?



Ključnega pomena za vzdrževanje nadzora nad dogodki je dokumentiranje in lastna organiziranost. Največkrat je dovolj že samo preprosto pisanje korakov, ki jih izvajamo. Ko je delo končano, pa si je potrebno vzeti čas in dokumentacijo tako oblikovati, da jo bo lahko uporabil še kdo. Ko je enkrat postopek evidentiran, lahko brez večjih težav take dogodke posredujemo svojim sodelavcem. S tem razbremenimo sebe, širimo znanje med sodelavci in ustvarjamo pozitivno klimo. Kar pa je še pomembnejše — organizacija na tak način ne postane odvisna od določene osebe.

Kaj pomeni obvladati področje?

Uspešnost sistemskega skrbnika (enako velja tudi za ostale) se lahko meri tudi po tem, kolikokrat so ga v trajanju njegovega dopusta klicali iz službe na njegov mobilni telefon. Morda ni ravno takoj razvidno, vendar je bolje, da v tem primeru skrbnik ne dobi nobenega klica. V nasprotnem primeru se pokaže, da je podjetje preveč naslonjeno na eno osebo, brez katere ne morejo »preživeti«. Znanje ni ustrezno porazdeljeno ali pa celotni sistem ni sistemsko postavljen in kot tak zahteva dosti intervencij.



Ker je okolje, v katerem sistemski skrbnik dela zelo široko in raznoliko, so tudi dogodki, s katerimi se srečuje, zelo različni. Eden od zelo preprostih in učinkovitih načinov, kako se dogodkov lotiti, nam razkrije sledeča slika.



Slika 4: Aktivnost skrbnika med reševanjem dogodkov

Vir: <http://video.google.com/videoplay?docid=7278397109952382318#>, (3.3.2011)

Slika nam razkrije, da če se dogodki pojavljajo zelo redko in so po težavnosti lahki, je najbolje, da se takim dogodkom preprosto posvetimo vedno znova od začetka. Dokumentacija nam v tem primeru vzame več časa kot pa rešitev sama. Tudi frekvenca pojavljanja nam tukaj ne prinese zelenih časovnih prihrankov. Primer za to bi bil vnos novega uporabnika.

Ravno nasprotno pa se dogaja, če imamo sicer lahke dogodke vsak dan na dnevnem redu. V tem primeru je smiselno razmišljati, da bi takšne postopke avtomatizirali. Največkrat se za avtomatizacijo skriva pisanje preprostih skript, ki se izvajajo v lupini računalnika ali znotraj programa, če nam ta to dopušča. Vseh dogodkov se sicer ne da avtomatizirati, vendar so preprosti dogodki vsekakor primerni za te postopke. Primer za to bi bilo izvajanje varnostnega kopiranja na tedenskem/dnevnem/ali celo krajšem obdobju. Pri tem se uporabljajo orodja, kot so *cron* in *at* v Linux okolju ali Razporejevalnik opravil (*Task Scheduler*) v Windows okolju.

Uporaba orodja *cron* v sistemu Linux

Želimo zapisati nalogo, ki nam bo vsak dan ob 21¹⁵ kopirala vsebino izbrane mape na drugo lokacijo sicer istega diska.

- Urejanje avtomatskega opravila izvedemo z ukazom `crontab -e`.
- Nato z urejevalnikom zapišemo naslednjo vrstico:
`15 21 * * * cp -R /home/uporabnik/pomembno /kopija/`

Pomen zapisa je sledeč:

```
* * * * * ukaz
| | | | |
| | | | - dan znotraj tedna [0 - 6] (0 pomeni nedeljo)
| | | - mesec [1 - 12]
| | - dan znotraj meseca [1 - 31]
| - ura [0 - 23]
- minuta [0-59]
```

* predstavlja vse možne vrednosti iz nabora

- Spremembo shranimo in avtomatska naloga se že izvaja.

Avtomatizacija dela lahko skrbnikom prihrani veliko časa in tudi virov. Izvajanje celotnega varnostnega kopiranja je zelo zahtevno opravilo za diskovni sistem. Če bi ga izvajali med delovnim časom, bi s tem upočasnili delo vsem ostalim uporabnikom. Tako pa lahko brez težav izvajamo take naloge, tudi če nas takrat ni v službi.

Poleg prihranka časa je morda pri avtomatizaciji opravil še pomembnejši podatek ta, da izvedba take naloge ni vprašljiva. Skrbniki so tako kot vsi ostali pozabljivi ljudje, pa še preobremenjeni so največkrat.



Na podoben način lahko razmišljamo tudi takrat, ko imamo sicer veliko dogodkov v kratkem obdobju, vendar so ti lahko zelo zahtevni. V takšnih situacijah je prav tako priporočljiva avtomatizacija, vendar je sedaj realizacija tega dosti težja. Tukaj nam preproste skripte ne morejo več pomagati in tako moramo sami razviti določena orodja, ki nam pomagajo pri reševanju dogodkov. Največkrat pa je še boljša rešitev nakup ustrezne specializirane rešitve. Primer bi lahko bil sistem CRM, ki nam pomaga beležiti, delegirati, dokumentirati, časovno in finančno spremljati incidente ter obveščati o aktivnostih pri njihovem odpravljanju.

Četrta možnost, kjer se redko pojavljajo sicer težki dogodki, pa se najlažje reši na tak način, da se vse rešene dogodke sproti dokumentira. Zbirka takšnih rešitev je lahko čez nekaj let izredno velikega pomena. Primer bi lahko bila zahteva za ponovno kreiranje certifikata, ker bo trenutnemu potekla veljavnost.

2.2 NALOGE IN POSTOPKI SISTEMSKEGA SKRBNIKA

Sedaj ko vemo, kako naj bi sistemski skrbniki začeli z delom, si pogledjmo še, kakšne so njihove tipične naloge oziroma s katerimi postopki se srečujejo. Vrste in obseg nalog ter postopkov so sicer močno odvisni od velikosti organizacije in področja, s čimer se ukvarja. Prav tako pa je potrebno upoštevati tudi število zaposlenih v informatiki, ugled, ki ga ta informatika ima pri zaposlenih in pri vodstvu ter splošno stanje odnosov med vodstvom in zaposlenimi ter med zaposlenimi samimi. V nobenem primeru pa ne moremo oziroma ne bi smeli naloge razvrščati po pomembnosti. Vse naloge so nujne za uspešno in predvsem varno delovanje celotnega sistema in s tem tudi organizacije. Kljub temu pa zaradi povsem praktičnih razlogov sistemski skrbniki naredijo prav to. Podlagi, na kateri to izvedejo, pa sta lahko dve. Prva je ta, da se oceni, kakšna škoda lahko nastane, če se nekega postopka ali naloge ne izvede. Večja kot je škoda, višje na prednostni lestvici se znajde naloga. Drugo pa je dejstvo, da so nekateri uporabniki pomembnejši od ostalih. Tu mislimo predvsem na vodstvo, ki je tisto, ki naše delo tudi ovrednoti in nam neposredno tudi reže kruh. Kakorkoli pa gledamo na prednosti, se pravilnost odločanja, kaj je najpomembnejše, povečuje z izkušnjami, ki jih skrbnik pridobiva z leti.

V našem primeru bomo zato podali sorazmerno širok seznam nalog in postopkov. Naslonili se bomo na večje organizacije, v katerih je več zaposlenih skrbnikov, se pa tudi v manjših organizacijah pojavlja podoben nabor aktivnosti. Naloge so razvrščene po sklopih, njihova frekvenca izvajanja pa je lahko dnevna, tedenska, mesečna, letna ali občasna.

2.2.1 Podpora uporabnikom

To je najvidnejša skrbnikova funkcija! Največja nevarnost, ki se tu lahko pojavi, je zapostavljanje drugih nalog zaradi te. Lahko pa tudi obratno! Če jih naštejemo, so to:

- Dodajanje/brisanje uporabnikov in spreminjanje njihovih pravic (avtorizacija).
- Zbiranje informacij o težavah, ki jih uporabniki imajo.
- Razvijanje novih rešitev, preizkušanje in uvajanje le-teh. Tu lahko nastopi težava, če te uporabniki začnejo prehitevati. Primer bi lahko bilo izvajanje marketinga preko Facebooka, kot skrbnik pa (še) ne veš, ali je iz varnostnega vidika to sprejemljivo.
- Vodenje evidenc o uporabniških računih, prijavih, dostopih do naprav, dostopih do datotek.
- Usposabljanje/izobraževanje uporabnikov za uporabo novih programskih rešitev, ki so bila/bodo nameščena.
- Priprava kratkih navodil za uporabnike.

- Pomoč uporabnikom pri raznih sistemskih vprašanjih ali situacijah, v katerih so se znašli. Kot primer bi podali povrnitev datotek iz varnostnega arhiva, raznih vprašanj v povezavi z varnostjo, omejitvami, ki jih imajo v povezavi s pošto in spletnimi storitvami (SPAM, zavrnitev pošte, lažni pošiljatelji, spletne prevare), okužbe z virusi. Tu gre predvsem za službene zadeve. Čisto nič pa ni narobe, če jim pomagamo tudi pri kakšni zasebni zadevi, saj s tem dvigujemo raven varnostne ozaveščenosti in dolgoročno s tem dvigujemo ozaveščenost tudi znotraj organizacije.

2.2.2 Vzdrževanje strojne in komunikacijske opreme

To je skoraj edino področje, kjer zaposleni lahko vidijo rezultate vašega dela in tudi glavna oprema se nahaja pri njih. Zato je dobro s pomočjo takih opravil vzdrževati odnos z zaposlenimi in prisluhniti njihovimi težavam in željam. Ta so:

- Nakupi nove strojne opreme ter prerazporejanje (računalniki, tiskalniki, komunikacijska oprema).
- Selitve uporabnikov iz enega prostora v drugega.
- Recikliranje/odpis stare opreme. Tu lahko prihaja tudi do odprodaje ali podaritve opreme.
- Vzdrževanje UPS naprav, sistemov za hlajenje strežniških sob, projekcijska oprema, kontrola pristopov, požarna varnost, IP telefonija.
- Nadzor nad kvaliteto izpisa in delovanja tiskalnikov. Tu se lahko pojavlja tudi vprašanje, kdo je zadolžen za nakupe potrošnega materiala.
- Vzdrževanje fizične in logične omrežne opreme — aktivna in pasivna oprema.
- Odpravljanje težave z internetnim ponudnikom. Tu so mišljeni predvsem dogodki ob izpadu internetne povezave ali v zvezi s kvaliteto te storitve.

2.2.3 Vzdrževanje programske opreme

Naloge, ki izvirajo iz tega področja, so vezane predvsem na preventivno vzdrževanje in zagotavljanje dobre kondicije posameznega računalnika. Tako pa se zagotavlja tudi zanesljivost celotnega sistema. Poleg tega pa sem prištevamo še odpravljanje raznih okužb in povsem nove postavitve operacijskih sistemov.

- Vzdrževanje normalnega delovanja operacijskega sistema (količina začasnih datotek, količina prostega prostora na disku, posodobitve sistema) in ostale programske opreme, ki jo organizacija dovoljuje nameščati (namestitve, nastavitve, nadgradnje, odstranjanje).
- Preprečevanje virusnih okužb in po potrebi čiščenje.
- Zagotavljanje podpore za nameščeno programsko opremo.
- Skrb za programsko opremo na strežnikih in servise, ki jih ti zagotavljajo.
- Pisanje skript za avtomatizacijo dela.

2.2.4 Vzdrževanje storitev

V to področje spadajo naloge iz naslova storitev, ki jih zagotavljamo za naše zaposlene. Predvsem gre tu za programsko opremo, ki skrbi za pošto, spletne strani, delovanje ERP sistema, baz podatkov. Želja je, da bi zagotovili zanesljivost delovanja sistema in odpornost na nepredvidljive dogodke znotraj organizacije.

- Poštni strežnik. Ta storitev predstavlja najpomembnejšo komunikacijsko povezavo v podjetju in jo uporabljajo vsi v organizaciji. V današnjem času je tu še poseben poudarek pri mobilnem dostopu.

- Spletni strežnik nam streže tako spletne strani (Internetne strani) kot tudi Intranetne in Extranetne strani in je pomemben predvsem zaradi ugleda, ki ga izgubimo, ko naše spletne strani niso dosegljive ali celo v primeru razobličanja (*Defacement*).
- Podatkovne baze.
- ERP poslovni sistem nam zagotavlja okolje za celovito upravljanje s poslovnimi procesi, ki jih imamo v organizaciji. To so največkrat proizvodnja, tehnologija, logistika, računovodstvo. ERP sistem je daleč najpomembnejši sistem znotraj vsake organizacije.
- Izvajanje varnostnega kopiranja.
- Pregled sistemskih virov je potrebno izvajati na redni mesečni osnovi. Taki podatki nam potem dajo vpogled, kakšen je trend rasti zasedenosti diska, obremenjenosti procesorja in podobni parametri. Na podlagi takih ugotovitev se potem izvajajo tudi korekcije sistema. Takim dejavnostim se mnogokrat reče tudi uglaševanje sistema (*Fine Tuning*).

2.2.5 Vzdrževanje varnosti

Področje varnosti je na žalost največkrat zanemarjeni del nalog skrbnika, vendar v resnici vsebuje nekatere naloge in postopke, ki so lahko ključnega pomena za preživetja organizacije v primeru hujših nesreč.

- Pregled skrbniških datotek (*Log files*).
- Spremljanje mrežnega prometa, stanja storitev, strežnikov in ostalih delovnih postaj pri sumljivih dostopih.
- Krpanje varnostnih lukenj na operacijskih sistemih in aplikacijah. Tu je mišljeno predvsem nameščanje varnostnih popravkov in razna penetracijska testiranja.
- Posodabljanje varnostnih nastavitvev (avtorizacije, nastavitve požarnih pregrad in IDS/IPS sistemov, raznih filtrov vsebin).
- Postavitev in spremljanje delovanja protivirusne zaščite na vseh napravah v organizaciji.
- Spremljanje in prilagajanje varnostnega sistema glede na trenutne razmere in namige/zahteve pri uporabnikih in vodstvu. To vključuje predvsem izvajanje varnostne politike.

2.2.6 Dokumentiranje aktivnosti in vzpostavitev/izvajanje/vzdrževanje ISMS dokumentacije

Izvajanje teh aktivnosti največkrat izkazuje, na kakšnem zrelostnem nivoju je informatika v podjetju/organizaciji. Vsa okolja, kjer je vsa zgodovina zabeležena samo v glavah sistemskih skrbnikov in kjer je delovanje sistema prepuščena intuiciji skrbnikov, pogostokrat tudi vodstvu, je dolgoročno obsojenih na propad. Morda je bil tak način dela pred 5 ali 10 leti še sprejemljiv, vendar bliskovit razvoj informatike kot vede, ki se vedno bolj približuje poslovnim procesom, zahteva popolnoma drugačen pristop. Tu nastopi sodelovanje in kot posledica tega je standardizacija ali vsaj določeni postopki dobre prakse, ki zaradi sodelovanja zahtevajo dokumentacijo.

- Zbirka rešenih dogodkov.
- Sezname kratkoročnih in srednjeročnih ciljev, usmeritev in dokumentacija na operativni ravni, kot so TODO liste in poslovne vizije.
- Krovni dokumenti informatike, ki zajemajo razne pojme, delitev dela, cilje informatike, vpetost v organizacijo.
- Sezname dobrin in groženj.
- Varnostna politika.
- Načrt za obnovo po nesrečah.

- Načrt neprekinjenega poslovanja.
- Sezname nameščene programske opreme in licenc.
- Sezname vseh uporabnikov in njihovih pravic.
- Ostala dokumentacija v povezavi s pridobivanjem/potrjevanjem raznih informacijskih varnostnih standardov kot so ISO 27001.
- Ostala dokumentacija v povezavi z vodenjem informatike v organizaciji, ki se nanaša na ITIL.
- Redno poročanje vodstvu o stanju/težavah/uspehih informatike.
- Razne pogodbe z zunanjimi izvajalci.
- Hranjenje namestitvenih medijev in vse pripadajoče dokumentacije v papirni ali elektronski obliki.
- Skrb za okolje, kot so odpadni papir, tonerji, kartuše, embalaža, skrb za manjšo porabo energije, papirja.

2.2.7 Dela, ki ne spadajo med naloge sistemskega skrbnika

Nalog, ki jih ima skrbnik sistema, je resnično veliko in razpon sposobnosti, ki naj bi jih imel, je širok. Zato imajo velika podjetja oziroma organizacije zaposlenih več skrbnikov. Težave pa nastanejo, če je skrbnik sam, ali je vodstvo nerazumevajoče. V takih okoljih pridobi skrbnik še dodatne naloge, ki pa naj ne bi spadale v službeno obveznost. Se pa pogosto to dogaja zato, ker so skrbniki največkrat zelo tehnične osebe in ki ne znajo reči ne.

Dela, ki naj jih skrbnik ne bi opravljal:

- Nedelovanje fotokopirnega stroja, FAX naprave ali avtomata za kavo oz. mnogokrat vse, kar je priključeno na elektriko.
- Skrb za spletne strani. Informatik ali skrbnik je zadolžen za infrastrukturo, ne pa za vsebino. Zanj naj bi skrbeli drugi strokovnjaki, kot so prodajne službe in službe, odgovorne za odnose z javnostjo ob pomoči skrbnikov, ki jim morajo zagotoviti možnost preprostega vnašanja vsebin.
- Izdelava elektronskih predstavitev za vodstvo.
- Skrb za direktorjev domači računalnik.

Razprava med kolegi

Če ima kateri od študentov izkušnje s sistemskim skrbništvom, naj svoje izkušnje predstavi ostalim. Doseči naj poizkuša, da se med kolegi razvije razprava o tem, kakšna dela skrbnik sistema opravlja oziroma kako se rešuje problem nepričakovanih (kriznih) dogodkov.



2.3 IZOBRAŽEVANJE SISTEMSKIH SKRBNIKOV

Od javno priznanih izobraževalnih programov je v Sloveniji neposredno namenjen področju sistemskega skrbništva samo višješolski program informatike. Ostali programi so ali preveč splošni ali pa namenjajo premalo ur. Vendar lahko vsak, ki želi delati ali poglobiti znanja na področju sistemskega skrbništva, osvoji posamezna področja tako, da se priuči s pomočjo svetovnega spleta in sodelavcev ali pa se udeleži posebnih strokovnih izobraževanj, ki se izvajajo z namenom pridobiti strokovnjake, za opravljanje nalog sistemskega skrbnika na specifični programski opremi. Na takih izobraževanjih zahtevajo določena predznanja.

Vodilni proizvajalci programske in strojne opreme so sčasoma ugotovili, da ni dovolj samo prodati izdelek. Mnogokrat je pomembnejše, da kupec zna rokovati z izdelki, saj se predvsem s tem izoblikuje zadovoljstvo z nakupom. Tako so za svoje izdelke izoblikovali posebne programe izobraževanja, ki se končajo s preverjanjem znanja in osvojitvijo določenega

certifikata. Ti so potem zagotovilo podjetju, da njihovi zaposleni obvladajo tehnologijo, zaposlenemu pa so lahko nagrada/priznanje, da mu podjetje zaupa, saj so po navadi taka izobraževanja zelo draga. Ne moremo pa spregledati dejstva, da lahko to pomeni tudi pomembno referenco pri iskanju zaposlitve.

V nadaljevanju si pogledjmo nekaj najpomembnejših oziroma najbolj obiskanih izpitov za sistemske skrbnike. Način in število izpitov se sicer iz leta v leto spreminja, je pa naslednji pregled dober vpogled v področje izobraževanja sistemskih skrbnikov na vseh nivojih.

MCSA (*Microsoft Certified Systems Administrator*): Naziv certifikata predstavlja stopnjo, ki jo po navadi pripisujemo sistemskemu skrbniku na srednji stopnji, kar pomeni med nekaj meseci in enim letom izkušenj na posameznih področjih. Certifikat je sestavljen iz 4 izpitov, od katerih sta dva določena, eden je iz področja namiznih operacijskih sistemov in eden je poljuben. Posamezni izpiti so vzeti iz nabora približno 40 izpitov. Vsak od teh izpitov pokriva eno ozko področje. Primeri: strežniško okolje, mrežne nastavitve, domensko okolje, SQL strežnik in podobne. Skrbniki začetniki začnejo z opravljanjem poljubnih izpitov oziroma tistih, iz katerih potrebujejo znanja na delovnem mestu, ki ga opravljajo. Po navadi je to namizni operacijski sistem. Ko opravijo dovolj izpitov ali dovolj pravih izpitov, pridobijo naziv MCSA in s tem nekako izkažejo svoj napredek na lestvici sistemskih skrbnikov. S prehodom na platformo Windows server 2008 se je uvedlo drugačne nazive, in sicer se en posamezen izpit sedaj imenuje MCTS, MCITP pa je naziv, ki se ga pridobi, če se opravi več določenih in izbirnih izpitov, vendar je število izpitov odvisno od nam zanimivih področij, iz katerih opravljamo izpite (<http://www.microsoft.com/learning/en/us/certification/view-by-name.aspx>, 7. 4. 2011).

MCSE (*Microsoft Certified Systems Engineer*): Naziv certifikata predstavlja stopnjo sistemskega skrbnika na najvišji ravni in to je arhitekt. Tu se pričakuje, da zna skrbnik sisteme tudi postavljati, ne samo vzdrževati. Stopnja izkušenj se tu ocenjuje med enim in dvema letoma na posameznem področju. Pristop je tu enak kot pri MCSA, le da je število izpitov večje, in sicer 7. Od tega so 4 iz področja komunikacij ter po eden iz področij namiznega operacijskega sistema, postavljanja sistemske arhitekture ter en poljuben izpit. S prehodom na platformo Windows server 2008 se je uvedlo nov naziv MCITP (<http://www.microsoft.com/learning/en/us/certification/view-by-name.aspx>, 7. 4. 2011).

RHCE (*Red Hat Certified Engineer*): Izpit je namenjen sistemskim skrbnikom, ki pri svojem delu uporabljajo Linux operacijske sisteme, ki jih je razvil Red Hat. Izpita sta dva, in sicer eden iz področja osnovne sistemske administracije in enega sestavljenega iz področij napredne sistemske administracije, kar vključuje tudi omrežne ter varnostne nastavitve (http://en.wikipedia.org/wiki/Red_Hat_Certified_Engineer#Red_Hat_Certified_Engineer_.28RHCE.29, 13. 4. 2011).

CCNA (*Cisco Certified Network Associate*): Je izpit, ki ga podeljuje podjetje Cisco in predstavlja certifikat, ki naj bi ga imeli vsi skrbniki računalniških mrež, seveda če uporabljajo strojno opremo podjetja Cisco. Izpit pokriva področja, kot so priklop, nastavitve, upravljanje delovanja in odprava težav na manjših in srednje velikih računalniških omrežjih. To v slovenskih razmerah pomeni, da se pokrije 99% vseh podjetij/organizacij. Do certifikata je ravno tako možno priti na več načinov (<http://en.wikipedia.org/wiki/CCNA>, 13. 4. 2011).

Znati ali ne znati?

En pristop na Microsoftov izpit (MCSA ali MCTS) stane približno 150\$, kar pa vključuje samo preverjanje znanja. Predhodno obiskovanje seminarja ni obvezno, je pa močno priporočljivo. Zato na spletnih straneh izvajalcev izobraževanj ugotovite koliko stanejo tečaji in seminarji v obliki laboratorijskih vaj in predavanj. Cena opravljanja izpita je v takih primerih po navadi kar vključena v ceno seminarja. Znanje torej ni poceni!

Primerjajte ceno posameznih izobraževanj in se pogovorite v kolikšnem času se taka investicija v znanje povrne.



2.4 ITIL (*Information Technology Infrastructure Library*)

Sistemske skrbniki so po navadi znotraj podjetja prepuščeni lastni presoji kaj je za njih najbolje in kakšne načine dela ter organizacijo bodo uporabljali. Vendar se lahko hitro zgodi, da takšen način razmišljanja hitro pripelje do nazadovanja razvoja informatike znotraj podjetja/organizacije in kot taka ne sledi spremembam v poslovnih procesih. Rešitev za takšne nevarnosti lahko najdemo v raznih standardih in priporočilih za organizacijo dela. Eden od takih je ITIL, ki si ga bomo sedaj na grobo pogledali.

ITIL v3 je v svoji zunanji podobi sestavljen iz petih knjig, od katerih vsaka pokriva eno področje znotraj celotne poti zagotavljanja podpore poslovnim procesom (http://en.wikipedia.org/wiki/Information_Technology_Infrastructure_Library, 12. 4. 2011).

Namreč informatiko je potrebno razumeti izključno kot servis vsem poslovnim procesom znotraj podjetja in nič drugega. Posledično to pomeni da se informatika prilagaja poslovanju in ne obratno. In sedaj da ni odločanje v preveliki meri prepuščeno vodji informatike ali sistemskemu skrbniku se poslužujemo raznih seznamov dobrih praks. ITIL ni tako nič drugega kot seznam zelo dobro razdelanih navodil kako organizirati delo da bodo učinki karseda veliki, poslovni procesi pa ne preveč moteni. Teh navodil se seveda lahko striktno držimo, lahko pa tudi ne.

Knjige, ki sestavljajo priporočila ITIL, so razdeljena na naslednja področja:

- Strategija storitev (*ITIL Service Strategy*) je osrednja knjiga, kjer se nahaja seznam vseh storitev, seznam vseh zahtev in finančni vidik zagotavljanja storitev.
- Načrtovanje storitev (*ITIL Service Design*) je tisti del, ki se najbolj dotika sistemskih skrbnikov in govori predvsem o upravljanju z incidenti in spremembami.
- Prenos storitev (*ITIL Service Transition*) govori o zahtevah poslovanja po prenosu storitev v operativno fazo.
- Operativna izvedba (*ITIL Service Operation*) govori o življenjskem ciklu storitve, ki je v operativni fazi.
- Redne izboljšave storitev (*ITIL Continual Service Improvement*) govori o načinu, kako upravljamo spremembe, ki se sčasoma pojavijo pri storitvah.

V tem učbeniku in namenu predmeta Vzdrževanje sistemske programske opreme je potrebno poudariti, da je priporočilo ITIL pisano za informatike. Torej tehnični vidik, ki naj bi ga sistemski skrbniki pokrivali, je samo košček celotne zgodbe, ki je zajet v knjigi Načrtovanje storitev. Zato si sedaj boljše oglejmo, kaj se v tej knjigi skriva:

- upravljanje z nivoji storitev,
- upravljanje tveganja,
- upravljanje kapacitet,
- upravljanje razpoložljivosti,
- upravljanje neprekinjenega poslovanja,
- in še druge.

Vsako od teh področij je upravljano s postopki, ki so zajeti v službi za podporo (*Service desk, Help desk, Call Center*):

- delo z incidenti (kakršnekoli napake, enotna točka prijave napak),
- delo z odpravljanjem težav (večje sistemske napake),
- delo s spremembami (sistemske spremembe v organizaciji),
- delo z izdajami (se navezuje na delo programerjev in razvojnih ekip).

Na spodnji sliki lahko vidimo zaslonsko sliko, kako izgleda primer uvajanja spremembe v okolje. Poleg zahteve, ki jo posreduje prijavitelj, lahko opazimo tudi, da je zahteva razporejena v določeno kategorijo in skrbnik kategorije zahtevo preda v izvajanje določeni osebi ali skupini, ki je usposobljena za pravilno delovanje tega področja (kategorije). V nadaljevanju lahko na spodnjem delu slike opazimo, katere aktivnosti je sistemski skrbnik izvedel, da je dosegel cilj, ki mu je bil zadan. Ko se izvaja takšno beleženje aktivnosti, si istočasno ustvarjamo sled našega dela, ki ga lahko kasneje uporabimo kot vir znanja za nove zaposlene, obračun delovnega naloga in obveščanje o trenutnem stanju pri realizaciji naloge. Če skrbnik ni zmore zanih nalog, lahko le-ta posreduje zahtevo svojemu sodelavcu, ki je izkušenejši ali ima višjo izobrazbo, naziv. Tako je brez truda in dodatnega komuniciranja lahko dosežena večnivojska podpora.

Poseben poudarek pri takem načinu dela je, da je nujno potrebno vsako aktivnost zabeležiti, saj se le tako gradi baza znanja. In tega pravila se morajo držati vsi zaposleni v informatiki. Še več, glavna težava je v tem, da morajo vsi uporabniki napake ali spremembe naročiti preko takega sistema. Informatikom je sedaj potrebno samo na nek način »prepovedati« opravljjanje kakršnih koli aktivnosti, če zahteva za to ni prej vnesena v sistem. Torej na tak način odpade sporočanje po telefonu, na hodniku, ob kavi in podobno, saj to pomeni lažje načrtovanje dela in razporejanja virov.

The screenshot shows the GLPI web interface in Internet Explorer. The main menu includes 'Inventar', 'Pomoč', 'Upravljanje', 'Orodja', 'Administracija', and 'Nastavitve'. The current view is 'Sledenje' (Tracking) for ticket 'Naročilo 376'. The ticket details are as follows:

Vir naročila:		Dodeljeno k:		Zadnja nadgradnja:	
Status:	V teku (dodeljeno)	Vir naročila:	Helpdesk	Skupaj trajanje:	1 Ura(e) 5 Minuta(e)
Prioriteta:	Srednje	Predmet:	Računalniki - VM CentOS - ...	Časovni stroški:	0.00
Kategorija:	05 Omrežje (M. Podbršček)	Tehnik:	Martin Bozic	Fiksna cena:	0.00
Uporabnik:	Roman Cuk	Skupina:	Skupina za omrežje in strežnike	Cena materiala:	0.00
Skupina:	----	Dobavitelj:	----	Skupna cena:	0.00

Additional information: 'Opombe po e-mailu: Da', 'E-mail naslov za opombe: roman.cuk@tsc.si'. A table of activities is shown below:

Datum	Opis	Trajanje	Načrtovanje	Avtor	Zasebno
15-04-2011 12:15	Težave z WinSCP in ključ. Ustvarjen direktorij	1 Ura(e)	Nobeden	Martin Bozic	Da
15-04-2011 09:40	Roman, prosim, če mi pošlješ javni ključ (kar po e-pošti). Program za generiranje SSH ključa: http://tartarus.org/~simon/putty-snapshots/x86/puttygen.exe WinSCP: http://winscp.net/eng/download.php	5 Minuta(e)	Nobeden	Martin Bozic	Da

Slika 5: Primer zaslonske slike Help desk sistema GLPI za področje podpore uporabnikom na TŠC Nova Gorica (v tem primeru delo s spremembami).

Uvajanje priporočil ITIL v podjetje je dolgotrajno opravilo in zelo utopično bi bilo razmišljati, da bi ga v celoti uvedli v enem zamahu. Najpreprosteje je realizirati del prve in druge knjige, kjer zajamemo sezname, finančne okvirje in upravljanje z incidenti. To pa je v grobem tudi tisto, kar sistemski skrbnik obvlada. Torej na tak način postopoma in po potrebi uvajamo priporočila, s tem pa posegamo v poslovne procese samo toliko, kot je nujno potrebno, in si istočasno ne delamo prevelikih stroškov.

Povzetek

Star pregovor pravi, da če lahko gre kaj narobe, bo narobe tudi šlo. V informatiki pa velja še dodatek, ki pravi, da težave bodo in če bodo, bodo vse istočasno na kupu. Obvladovanje časa je tako za notranje zadovoljstvo sistemskega skrbnika ključnega pomena. Skrbnik ne more storiti vsega naenkrat in nemogoče je, da ne bo kateri od uporabnikov moral čakati na rešitev. Pritiski, ki jih uporabniki izvajajo nad skrbniki, so sicer upravičeni, vendar s tem močno vplivajo na aktivnosti, ki jih je skrbnik načrtoval. To pa pomeni, da rokov, ki si jih je zadal, ne bo mogel izpolniti, s tem pa se bo nezadovoljstvo pri zaposlenih in tudi pri njem še povečalo.

Da se čim bolj izognemo opisanim dogodkom, je potrebno dvoje. Uvedba sistema spremljanja dogodkov, ki nam beleži aktivnosti, porabljen čas in istočasno omogoča načrtovanje porabe časa in virov. Drugi način pa je izobraževanje, s čimer se poveča hitrost reševanja incidentov ali razbremenitev sodelavcev. Na tak način lahko povečamo število usposobljenih skrbnikov in možnost delitve dela se s tem poveča. Pri prvem načinu se spogledujemo s katerim od standardov oziroma s seznamami dobrih praks (ITIL), v drugem primeru pa se poslužujemo izobraževanj, kot so MCSA, MSCE, RHCE ali CCNA.

Vprašanja in naloge

- 1) Kako se imenuje začetna stopnja, na kateri sistemski skrbnik začne svojo kariero?
- 2) Na kakšen način bi se kot sistemski skrbnik lotili težav, ki so na eni strani označene kot lahke ali težke, po drugi strani pa so označene kot pogoste ali redke? Poizkusite najti ustrezne primere.
- 3) S kakšno metodo bi si pomagali pri vodenju dnevnih opravil, ki vas čakajo tistega dne? Ali je metoda uporabna tudi v vsakdanjem življenju?
- 4) Naštejte glavna področja, na katerih mora biti sistemski skrbnik aktiven.
- 5) Naštejte nekaj dnevnih, tedenskih in mesečnih opravil ter nekaj opravil, ki niso vezana na koledar.
- 6) Kaj je glavni namen zbirke ITIL in ali se ga smatra kot standard?
- 7) Kakšne naj bi bile aktivnosti oziroma postopek ob prijavi incidenta v sistem uporabniške podpore?

3 UPRAVLJANJE PROGRAMSKE OPREME

Cilji poglavja

V poglavju sledimo izzivom, s katerimi se skrbniki srečujejo pri nameščanju in upravljanju programske opreme, od izbire programske opreme do njene uporabe in končne odstranitve. Spoznamo različne načine nameščanja tako v grafičnih okoljih kot tudi ukazni lupini.

Za učinkovito spremljanje vsebin je potrebno znanje iz uporabe različnih operacijskih sistemov (uporabniška raven), klasifikacij programske opreme, brskanja v spletu in osnov programiranja..

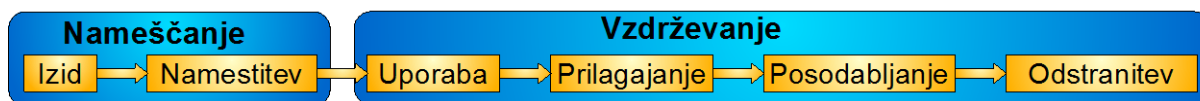


Izziv

Pri informatiku v podjetju se je oglasil direktor in izrazil pričakovanje, da bo informatik v naslednjem dnevu na vseh 50 računalnikov v podjetju namestil novo različico programske opreme, ki jo podjetje uporablja za svoje delo. Kako naj se informatik spopade z izzivom?



Če imamo samo nekaj računalnikov, ima vprašanje programske opreme manjšo težo, kot če upravljamo nekaj deset ali celo sto računalniških sistemov (http://en.wikipedia.org/wiki/Software_deployment, 28. 4. 2011). Wikipedia opredeljuje uvajanje programske opreme (*Software Deployment*) kot nabor vseh aktivnosti, s katerimi uporabnikom pripravimo programsko opremo za rabo. Proces uvajanja je sestavljen iz več faz, ki jih lahko razdelimo med dve večji fazi.



Slika 6: Faze uvajanja programske opreme

3.1 NAMEŠČANJE IN POSODABLJANJE PROGRAMSKE OPREME

Uvajanje programske opreme se prične z njenim izidom in dostopnostjo na trgu. Ko podjetje programsko opremo pridobi, lahko prične razmišljati o njeni namestitvi. Programsko opremo lahko nameščamo (in posodabljam) na raznolike načine. Vsak način ima svoje prednosti in slabosti. Nameščamo in posodabljam lahko:

- iz izvorne kode, iz binarne kode;
- z enostavnim kopiranjem, z namestitvenim programom ali iz »shrambe«;
- na daljavo ali v živo;
- interaktivno ali brez interakcij — avtomatizirano;
- centralizirano ali decentralizirano.

Hitreje se opravi nameščanje programske opreme iz binarne kode, saj je pri nameščanju iz izvorne kode potrebno na sistemu vzpostaviti razvojno okolje. Upoštevajoč veliko število programskih jezikov in različnih ogrodij za programiranje, je lahko to opravilo zelo mučno. V naslednjih primerih se namestitev vseeno opravlja iz izvorne kode:

- dostop do najnovjših ali razvojnih različic programske opreme,
- prilagajanje programske opreme posebnim zahtevam podjetja,
- optimizacija programske opreme za izbrani sistem.

Če potrebujemo prilagajanje programske opreme, podjetje izdelava svoje ali pa uporabi prilagoditve drugih. Tovrstne prilagoditve imenujemo **zaplata** (*Patch*). Z ustrežno

programsko opremo se izvorno kodo dopolni oziroma spremeni in nato prevede v binarno ter uporabi.

Večkrat se programska oprema distribuira v samoizvedljivi obliki. Označuje se tudi kot **prenosljiva**. Glavna prednost je torej v prenosljivosti. Tako programsko opremo je enostavno namestiti in posodobljati, saj se enostavno stara verzija prepíše z novejšo. Slabost je v običajno slabši odzivnosti in manjši uporabniški prijaznosti predvsem pri zaganjanju. Ker ni namestitvenega programa, ni vnosov v menijih znotraj raziskovalcev in drugih aplikacij. Tako tudi ni povezovanja različnih aplikacij med seboj. Prenosne aplikacije ne zapisujejo ničesar v sistemske imenike, ampak tipično le v imenik, od koder se jih zaganja. Ker vsi programi niso prilagojeni takemu izvajanju, včasih uporabljajo poseben zagonski program, ki pred zagonom aplikacije skopirajo nastavitve in datoteke na gostujoči sistem in jih po zaključku dela ponovno premaknejo v imenik programa (http://en.wikipedia.org/wiki/Portable_application, 28. 5. 2011). Na spletu najdemo že pripravljene aplikacije za prenos in takojšnjo uporabo (<http://portableapps.com/>).

Posebno poglavje med prenosljivo programsko opremo predstavljajo programi, ki za svoje izvajanje potrebujejo ustrezno izvajalno okolje. Taki programi se lahko brez ponovnega prevajanja poganjajo na vseh sistemih, za katere obstaja izvajalno okolje. Izpostavimo lahko C# in javanske programe. C# programi se poganjajo v .NET ogrodju (.NET *framework*). Microsoft je .NET ogrodje razvil za operacijske sisteme Windows, odprtokodna skupnost pa ogrodje vzpostavlja še za operacijske sisteme Linux in Mac OS X (<http://www.monoproject.com>, 2. 5. 2011). Javanski programi za svoje delovanje potrebujejo javansko izvajalno okolje (JRE – *Java Runtime Environment*). To je dosegljivo za večino operacijskih sistemov.

Preizkus

Uporabite spletni iskalnik in pobrsajte med zadetki iskalnega niza “portable software”. Našli boste veliko prenosnih aplikacij. Izberite poljubno in si jo prenesite. Hkrati pridobite namestitveni program za isto aplikacijo. Primerjajte odzivnost obeh. Kakšne razlike še opazite pri (daljši) uporabi?



V poglavju o virtualizaciji spoznamo koncept virtualizacije aplikacij, ki ima isti cilj – dosego prenosljivosti aplikacij, vendar je pristop drugačen. Pri virtualizaciji gre za uvedbo nove plasti, ki prestreže klice aplikacije in jih ustrezno preusmerja.

Zaradi omenjenih pomanjkljivosti se še vedno večinsko uporablja namestitvene programe, ki prenesejo programske datoteke v ciljne imenike in opravljajo nastavljanje na sistemu (pišejo v register v Windows operacijskih sistemih oziroma konfiguracijske datoteke na Linux sistemih). V preteklosti se je uporabljalo različne namestitvene programe, v zadnjem času pa prevladujejo namestitvene rešitve (<http://support.microsoft.com/kb/310598>, 2. 5. 2011), ki:

- povrnejo prejšnje stanje v primeru napake med nameščanjem,
- razrešujejo konflikte med različno programsko opremo,
- prepoznajo in popravljajo poškodovano programsko opremo,
- znajo zanesljivo odstraniti programsko opremo (pustijo npr. na sistemu knjižnice, ki jih uporabljajo drugi programi),
- podpirajo namestitev brez interakcij skrbnika (*unattended*),
- ob nameščanju, posodobljanju in odstranjevanju programske opreme ustrezno namestijo, posodobijo oziroma odstranijo tudi ustrezne knjižnice ipd. (http://en.wikipedia.org/wiki/Advanced_Packaging_Tool, 2. 5. 2011),
- podpirajo nameščanje z uporabo grafičnega vmesnika ali iz ukazne vrstice.

V okolju Windows tovrstni paketi nosijo končnico .msi, v linux okoljih pa .deb ali .rpm.

3.2 AVTOMATIZACIJA UPRAVLJANJA S PROGRAMSKO OPREMO

V primeru vzdrževanja večjega števila sistemov je nujno potrebno vzpostaviti čim bolj avtomatizirano upravljanje. Na trgu je prisotnih več rešitev, ki olajšujejo upravljanje na osnovi različnih konceptov. Nekateri koncepti se dopolnjujejo, nekateri pa tudi izključujejo.

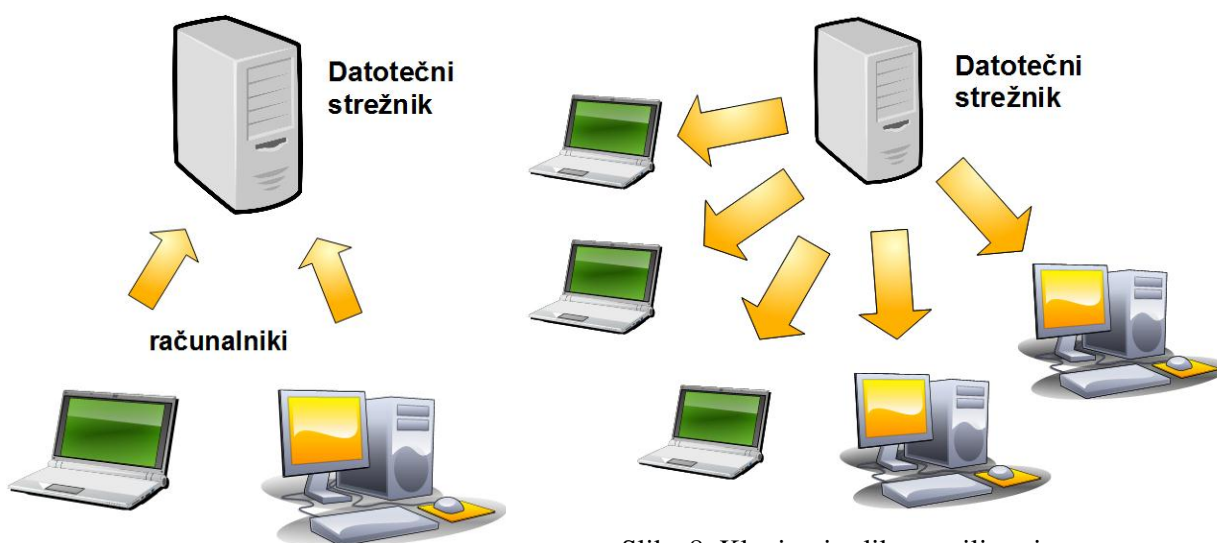
Iz prakse

Pri skrbniku se je oglasil računovodja in potožil, da dela njegova računalnik vsak dan počasneje, čeprav ni na njem vršil nobenih sprememb. A naj ne bi računalnik delal enako kot na začetku uporabe? Skrbnik je opravil pregled sistema in ugotovil, da se razen popravkov na sistemu res ni nameščalo ničesar. Kako naj skrbnik vzpostavi zeleno stanje?



Vzdrževanje je mogoče precej zmanjšati z uporabo rešitev za **zamrznitev stanja**. Strojno ali programsko osnovane rešitve omogočajo vzpostavitev začetnega stanja ob vsakem zagonu računalniškega sistema ne glede na aktivnosti uporabnika. Na sistemu se izključi samodejno posodabljanje. Razdelki s podatki uporabnikov se izločijo iz sistema zamrzovanja. Če potrebujemo spremembo nastavitve ali programske opreme, je potrebno najprej izključiti sistem zamrzovanja, opraviti namestitev in nato sistem zamrzovanja ponovno vključiti.

Začetno stanje lahko vzdržujemo tudi z uporabo **kloniranja sistemov**. Klonirati je mogoče obstoječi sistem ali pa sliko sistema. V drugem primeru se najprej postavi vzorčni sistem, njegova slika pa se ustvari na datotečnem strežniku. V drugem koraku se slika prenese na vse sisteme. Sliko je potrebno vzpostaviti za vsak tip sistemov posebej. Ob težavah se na težavnem sistemu enostavno vzpostavi začetno stanje. Rešitev se lahko uporablja v kombinaciji z zamrzovanjem.



Slika 8: Kloniranje slike na ciljne sisteme

Slika 7: Ustvarjanje slik novo vzpostavljenih sistemov

Razmislek

Pisarniški paket morate namestiti na več različnih sistemov. Na vseh sistemih bi radi namestili enake komponente sistema. A je res potrebno na vsakem sistemu klikati in čakati na znane odzive?



Avtomatizirati je mogoče tudi proces vzpostavljanja začetnega sistema. Če želimo nek program na enak način namestiti na več sistemov, lahko namestitev opravimo brez interakcij. Nekatera programska oprema omogoča nameščanje z uporabo stikal oziroma datoteke, ki

vsebuje odgovore na vprašanja, ki bi jih sicer namestitveni program postavil skrbniku. V spletu lahko najdemo tudi prosto dostopno programsko opremo, ki omogoča izdelavo takih neinteraktivnih namestitev kot tudi veliko že pripravljenih skript, ki tako namestitev poganjajo (<http://www.opsi.org/features/>, 13. 5. 2011).

V največjih sistemih se vzpostavlja centralizirano vodeno oddaljeno upravljanje s programsko opremo na vseh sistemih v omrežju. V okoljih, kjer je to potrebno, je upravljanje lahko vezano tudi na pravice uporabnika. V tem primeru se nameščanje in posodabljanje programske opreme na delovni postaji, ki je vključena v omrežje, opravlja na podlagi varnostnih politik za izbrano delovno mesto.

V Windows okoljih je smiselno razmišljati o vključevanju delovnih postaj v domeno, kjer se potem upravljanje programske opreme opravlja na osnovi Group Policy Objects (GPO). GPO so sestavni del aktivnega imenika in določajo cel nabor zmožnosti, od nastavitve spletnega brskalnika do nadzora nad delovanjem posameznih aplikacij, varnostnih nastavitvev ipd. GPO podpira upravljanje s programsko opremo na osnovi uporabnika ali na osnovi računalniškega sistema (Ruest, 2005).

Predpogoj za dobro upravljanje s programsko opremo je zgrajena baza strojne in programske opreme (*Inventory*) ter licenc. Upravljanje na daljavo vključuje:

- avtomatizirano nameščanje operacijskega sistema,
- avtomatizirano nameščanje in posodabljanje programske opreme.

Razmislek

Kdo je tisti, ki v podjetju odloči, katero programsko opremo se bo uporabljalo? Kdo odloča, kako se bo upravljalo s programsko opremo? Kdo s programsko opremo dejansko upravlja? Kje so odgovornosti zapisane?



Ne glede na uporabljane rešitve je vedno potrebno postopati po naslednjih korakih (Bierds, 2004):

- vzpostavitev ekipe, ki bo upravljala s programsko opremo, in vključuje odgovorne, inženirje in tehnike ter njihove vloge;
- pregled dokumentacije in drugih virov o programski opremi;
- sestava načrta za upravljanje;
- vzpostavitev razmerij z dobavitelji;
- vzpostavitev in predstavitev načrtov za upravljanje;
- izvajanje upravljanja v skladu z načrti.

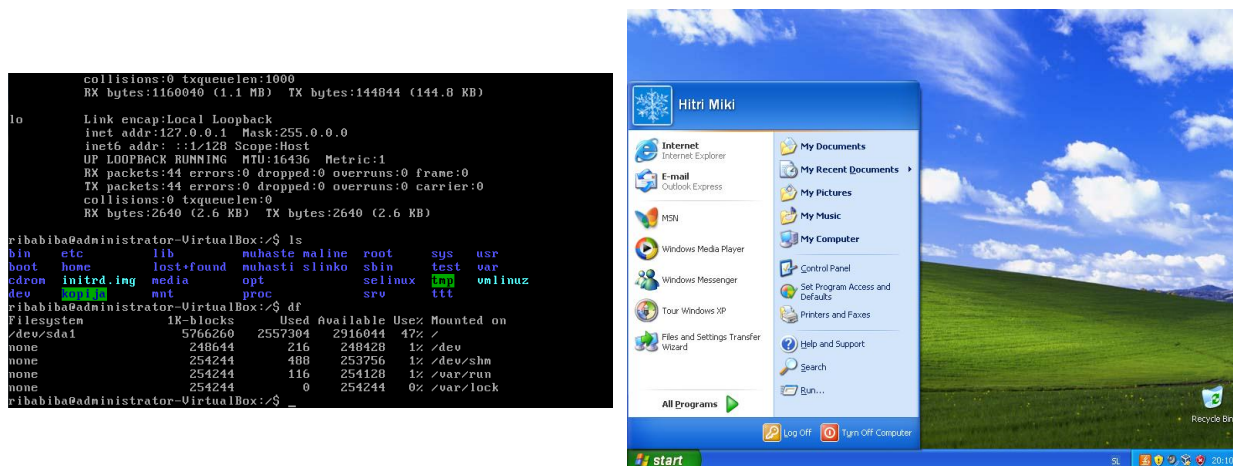
3.3 SKRIPTNI JEZIKI

V predhodnem poglavju smo si ogledali, kakšni so načini za obvladovanje programske opreme, ki je nameščena na velikem številu delovnih postaj, ali pa je okolje, v katerem te postaje tečejo, zelo podvrženo poškodbam programske opreme. Primer za to bi bilo šolsko okolje, kjer se lahko tedensko v eni računalniški učilnici zvrsti več profesorjev in vsak poučuje drug predmet in zato potrebuje vsak mesec kakšen drug program. Ravno tako tudi študentje pri izvajanju svojih nalog nameščajo razno opremo. Ta se sčasoma nabere in začne močno vplivati na hitrosti in način odzivanja ter delovanja samega sistema.

V tem poglavju pa se osredotočimo na avtomatizacijo, ki jo lahko izkoristi sistemski skrbnik pri svojem rednem vsakodnevem delu. Določena opravila, ki se pogosto izvajajo, je največkrat možno avtomatizirati. Tu si bomo sedaj pogledali, kakšna okolja so primerna za avtomatizacijo.

3.3.1 Tekstovni grafični vmesniki

Poznamo dve vrsti uporabniških vmesnikov. Prvi je uporabniku prijazen grafični vmesnik (*Graphical User Interface*), ki je zelo intuitiven in preprost za uporabo. Vendar je z našega stališča zelo zapleten za uvajanje avtomatiziranih opravil. Kljub temu da obstajajo makro jeziki (*Macro Language*), ki nam omogočajo določeno avtomatizacijo, pa ti jeziki niso niti približno tako razširjeni in izrazno močni, kot so skriptni jeziki (*Scripting Language*) na tekstovnih uporabniških vmesnikih (*Text User Interface*, *Command Line Interface*), ali drugače rečeno, uporabniška lupina (*User Shell*) (http://en.wikipedia.org/wiki/Scripting_language, 11. 5. 2011). Res da so tekstovni vmesniki nekako stvar zgodovine, vendar so zaradi izredno lepe in učinkovite komunikacije z uporabniki ter avtomatizacije opravil še vedno prisotni in zaželeni. Še več, tudi tisti operacijski sistemi za resno poslovno uporabo, ki so v zgodovini stavili na grafični uporabniški vmesnik, so začeli ponovno vgrajevati in posodabljati tekstovne vmesnike (<http://windows-powershell.software.informer.com/wiki/>, 8. 5. 2011). Na spodnji sliki lahko vidimo grafični izgled dveh različnih uporabniških vmesnikov.



Slika 9: Primer tekstovnega uporabniškega vmesnika in grafičnega uporabniškega vmesnika

Vsi sodobni operacijski sistemi imajo privzeto nameščen grafični uporabniški vmesnik. To velja tako za namizne postavitve kot strežniška okolja. Kljub uporabi grafičnega vmesnika ravno tako omogočajo tudi uporabo tekstovnega vmesnika, ki se ga najpogosteje uporablja v okenskem načinu. Na ta način sta združena tako prijaznost do uporabnikov kot tudi možnost izvajanja različnih avtomatiziranih opravil.

Diskusija med študenti

Na naslovu http://www.softpanorama.org/OFM/gui_vs_command_line.shtml, (7. 5. 2011) najdete zbirko nekaterih povzetkov na temo primerjave »grafično in tekstovno« iz raznih forumov. Med študenti naj se ustvari diskusija na to temo, vendar je potrebno vsako stališče tudi utemeljiti. Vsak naj vidi na obeh področjih pozitivne in negativne lastnosti. Še posebej komentirajte naslednje trditve:

- GUI je preprost za uporabo in zato je prijazen do uporabnika.
- GUI poneumlja uporabnike.
- CLI nudi uporabniku večji nadzor in več možnosti.
- CLI je zgodovina.
- GUI je napredek v primerjavi z CLI.
- GUI predstavlja veliko varnostno grožnjo.



- Uporaba GUI je hitrejša, saj je izbira ikon in elementov hitrejša kot pomnjenje in tipkanje ukazov in parametrov.
- Uporaba CLI je hitrejša, saj potrebujemo samo tipkovnico in nam ni potrebno premikati miške, uporabljati drsnikov in z vsako novo verzijo ponovno iskati ikon.
- GUI porabi veliko sistemskih virov.
- Sodobni računalniki so zmogljivi in imajo dovolj pomnilnika, tako da vidik porabe sistemskih virov ni več pomemben.

Nasprotno od pričakovanega pa se na strežniškem okolju v precejšnji meri pojavlja uporaba samo tekstovnega uporabniškega vmesnika. Tu so mišljene predvsem postavitve velikih sistemov iz družine Unix in Linux. V manjših postavitvah in v Windows okoljih pa ravno tako kot na namiznih rešitvah srečujemo oba vmesnika. Razlogi za nenameščanje GUI so z vidika sistemske izkoriščenosti virov povsem razumljivi, saj grafično okolje zahteva velik del pomnilnika in tudi večjo pasovno širino komunikacijske povezave v primeru oddaljenega dostopa na namizje. Drugi vidik nameščanja samo tekstovnega vmesnika pa je bolj subjektivne narave. Mnogi izkušeni sistemski skrbniki trdijo, da je delo v tekstovnem načinu hitreje od dela v grafičnem. Nasprotno pa se skrbniki začetniki s tem ne strinjajo in raje prisegajo na grafično okolje, ki jim s svojo grafično predstavitvijo vseh možnih »izbir« olajša delo. Tako je izbira uporabniškega vmesnika največkrat odvisna predvsem od izkušenj sistemskega skrbnika, vrste nalog, ki na strežniku tečejo, ter raznih odločitev proizvajalcev programske opreme.

Za dobro voljo

Na operacijskem sistemu Ubuntu v grafičnem okolju pritisnite kombinacijo tipk Ctrl+Alt+F1. Kaj se zgodi? Prijavite se z vašim uporabniškim računom. Poiščite, kako priti nazaj do grafičnega okolja. (Namig: poizkušajte tipke med F1 in F8).



V nadaljevanju se osredotočimo na tekstovne uporabniške vmesnike (TUI), za katere se največkrat uporablja ime Uporabniška lupina (*User Shell*) ali Ukazna vrstica (*Command Line Interface – CLI*). Tu gre za preprost koncept komunikacije z računalnikom, kjer uporabnik poda ukaz kot niz znakov, računalnik oziroma vmesnik pa odgovori z odgovorom v obliki niza znakov ali daljšega teksta.

Skozi zgodovino razvoja operacijskih sistemov so se razvile in se še razvijajo mnoge uporabniške lupine, ki se po svoji funkciji ne razlikujejo. Razlike se pojavljajo predvsem v ozadju delovanja same lupine, cevljenju ukazov (*Pipelining*), drugačnem naboru ukazov ter sintakse in nekoliko tudi v samem vizualnem izgledu ter pomoči, ki jo nudi uporabniku (http://en.wikipedia.org/wiki/Comparison_of_command_shells, 8. 5. 2011). Kljub pestrosti izbire med lupinami pa bomo omenili le dve, in sicer tisti, ki sta najbolj razširjeni. To sta Bash (Linux) in PowerShell (Windows). Ostale lupine, kot so csh, ksh, ash in večina ostalih pa se danes pojavljajo predvsem v sistemih, ki izhajajo iz družine BSD in niso pogoste.

Bash

Uporabniška lupina Bash je bila razvita leta 1989 in se zgleduje po predhodnih različicah lupin sh, csh in ksh in se od njih razlikuje samo v nekaterih podrobnostih ([http://en.wikipedia.org/wiki/Bash_\(Unix_shell\)](http://en.wikipedia.org/wiki/Bash_(Unix_shell)), 8. 5. 2011). Nabor ukazov je vzet iz lupine sh in kasneje razširjen. Sintaksa ukazov izhaja iz lupine ksh, uporabniška prijaznost, kot so zgodovina ukazov, dopolnjevanje imen in map, nekatere spremenljivke okolja ter možnost zamenjave ukazov, pa prihaja iz lupine csh. Bash lupina je danes prisotna v vseh distribucijah Linuxa in Mac OS X. Grafični prikaz je podan na zgornji sliki.

Windows PowerShell

Windows PowerShell je bil predstavljen leta 2006 in je nekako naslednik starejšega `command.com` in `cmd.exe` (<http://windows-powershell.software.informer.com/wiki/>, 8. 5. 2011). Vendar tu ne gre zgolj za novo preobleko, ampak predvsem za popolnoma nov koncept izvedbe uporabniške lupine. Deluje na objektnem nivoju in je sposobno interakcij z okoljem .NET. To dejstvo naredi lupino izredno zanimivo za izvajanje avtomatizacije in upravljanja, saj lahko nabor ukazov, ki jih lupina pozna, širimo tudi z lastnimi ukazi, ki jim pravimo Cmdlets. Osnovni ukazi in sintaksa so vzeti iz starejših Microsoftovih uporabniških lupin in iz okolja Bash. Nekaj osnovnih ukazov lahko vidimo v spodnji tabeli.

Tabela 2: Primerjava osnovnih ukazov Bash in Windows PowerShell

Windows PowerShell Cmdlet ali (okrajšava)	Bash	Opis
Get-Location (gl, pwd)	pwd	Prikaži trenutni položaj datotečnega kazalca
Set-Location (sl, cd, chdir)	cd, chdir	Zamenjaj trenutno pozicijo značke
Remove-Item (ri, del, erase, rmdir, rd, rm)	rm, rmdir	Briši datoteko ali direktorij
Copy-Item (cpi, copy, cp)	cp	Kopiraj datoteko ali direktorij
Get-Process (gps, ps)	ps	Prikaži seznam vseh procesov
Get-Help (help, man)	man	Pomoč pri delu

(Vir: <http://windows-powershell.software.informer.com/wiki/>, 8. 5. 2011)

Paketna datoteka (*Batch file – Windows ali Shell Script – Unix/Linux*)

Če večkrat ponavljamo iste ukaze v enakem zaporedju in z enakimi parametri, lahko celotno zaporedje ukazov shranimo v datoteko, ki ji potem rečemo skripta (http://en.wikipedia.org/wiki/Shell_script, 12. 5. 2011) oziroma paketna datoteka (http://en.wikipedia.org/wiki/Batch_language, 12. 5. 2011). Tipična končnica datotek je tako `.ps1`, `.bat` ali `.cmd` v Windows okolju ter `.sh` v Unix/Linux okolju (končnica je lahko tudi izpuščena). Poleg uporabe ukazov, ki jih lupina razume, lahko v paketni datoteki uporabimo tudi vejitve (if stavki in zanke), prirejanje vrednosti spremenljivkam ter uporabo parametrov za prenos vrednosti v izvajalno okolje.

Iz prakse

Primer uporabe skripte paketne datoteke za varnostno shranjevanje. Datoteko shranite kot `kopija.sh` in jo poženete s `./kopija.sh`. Za pravilno delovanje morate uporabljati bash lupino, sicer morate popraviti prvo vrstico.

```
#!/bin/bash
OF=/var/my-backup-$(date +%Y%m%d).tgz
tar -czf $OF /home/me/
```

Na naslovu <http://www.faqs.org/docs/Linux-HOWTO/Bash-Prog-Intro-HOWTO.html> najdete preprost vodič za delo s paketnimi datotekami. Razmislite, kako bi zgornji primer razširili tako, da namesto mape `»/home/me«` uporabnik sam pove, katero mapo bo izbral `»/home/izbor«`.



Skriptni jeziki

Skriptni jeziki so nadmnožica paketnih obdelav (http://en.wikipedia.org/wiki/Scripting_language, 11. 5. 2011). V zgodovini se je uporaba skriptnih jezikov širila in razvijala. Začelo se je z osnovnimi ukazi, ki jih je lupina poznala. Temu načinu dela je potem sledila razširitev nad tok izvajanja skripte in uporabo spremenljivk (<http://www.faqs.org/docs/Linux-HOWTO/Bash-Prog-Intro-HOWTO.html>, 12. 5. 2011). Nadgradnja paketne obdelave pa je razširitev s programskimi jeziki. Tu največkrat ne gre za

neposredno uporabo sintakse programskega jezika, ampak za približek le-tega. Tisto, kar je najpomembnejše, je to, da skripta za svoje izvajanje na potrebuje prevedenega »programa«, ampak se le-ta izvaja v izvajalnem okolju. Tako se skripta med samim izvajanjem interpretira ukaz za ukazom. Prednost takega načina izvajanja je, da postanejo skripte neodvisne od strojne in programske platforme. Torej postanejo povsem prenosljive med računalniškimi sistemi, vendar pa za izvajanje potrebujemo izvajalno okolje, katerega moramo najprej namestiti, če ga še nimamo, (<http://www.sthurlow.com/python/>, 14. 5. 2011). Primeri za sodobne skriptne jezike bi bili: PHP, ASP, JavaScript, VBScript, awk, Perl, Python, Ruby.

Povzetek

Proces uvajanja programske opreme sestavljata fazi nameščanja in vzdrževanja. Nameščanje se lahko opravlja iz izvorne ali binarne kode, s kopiranjem ali z namestitvenim programom, na daljavo ali v živo, z ali brez interakcije, v večjih sistemih centralizirano. Včasih je želja prosto dostopno programsko opremo dopolniti, uporabijo se zaplate. Programska oprema je lahko tudi prenosljiva, kar pomeni, da se distribuira v samoizvedljivi obliki. Uporaba novejših namestitvenih programov prinaša številne prednosti, predvsem možnost razreševanja konfliktov, prepoznavanje in popravljanje poškodovane programske opreme, prepoznavanje in nameščanje manjkajočih knjižnic idr. Z večanjem števila sistemov se večja potreba po avtomatizaciji upravljanja. Količino upravljanja zmanjšujejo zmožnosti zamrzovanja stanja, kloniranje sistemov, nameščanje brez interakcij in centralizirano vodeno oddaljeno upravljanje. V zadnjem primeru je nameščanje lahko odvisno od vloge, ki jo uporabnik ima, in ne od računalnika, ki ga uporablja.

Avtomatizacija opravil, ki jih sistemski skrbnik dnevno izvaja, je še ena od možnih avtomatizacij upravljanja. Tu gre predvsem za uporabo skriptnih jezikov, ki se izvajajo v tekstovnem uporabniškem vmesniku.

Vprašanja in naloge

- 1) Kdaj je smiselno razmišljati o avtomatiziranju upravljanja s programsko opremo?
- 2) V podjetju želite uvesti novo programsko opremo. Kako boste postopali?
- 3) Kdaj je smiselno programsko opremo namestiti iz izvorne kode? Kakšne težave lahko pričakujemo v primerjavi s programsko opremo, ki je nameščena z uporabo paketov?
- 4) Kaj je to prenosljiva programska oprema?
- 5) Kakšne so prednosti uporabe programske opreme, ki se namesti z ustreznim namestitvenim programom?
- 6) Kaj je to zamrznitev stanja? Navedite primer, kjer bi bila uporaba te metode smiselna, in primer, kjer to ne bi bilo dobro.
- 7) Ali lahko kloniramo sliko na sistem, ki ni enakega tipa kot izvorni sistem, na katerem je bila slika ustvarjena?
- 8) Kaj je to namestitev brez interakcije?
- 9) Opišite razlike med grafičnim in tekstovnim uporabniškim vmesnikom. Nakažite razloge, zakaj je katero od teh okolij uporabnejše.
- 10) Napišite skripto, ki bo v poljubnem skriptnem jeziku prebrala datoteko in jo izpisala na zaslon.
- 11) Spletni brskalnik Firefox je primer programske opreme, ki se lahko izvaja na različnih operacijskih sistemih. Opišite postopke nameščanja v Windows in Linux okolju in ju primerjajte. Kaj so prednosti oziroma slabosti posameznih rešitev? Katere so možnosti nameščanja brez interakcije?

4 VARNE SPLETNE STORITVE

Cilji poglavja

V nadaljevanju je podan pregled sistemske programske opreme s stališča različnih spletnih storitev. V prvem delu se srečamo s kompleksnostjo področja varnosti in se v nadaljevanju spustimo na vzpostavljanje varnih spletnih storitev.

Za učinkovito spremljanje vsebin je potrebno znanje iz upravljanja računalniških sistemov (upravljanje uporabniških računov, nameščanje sistemske programske opreme) in poznavanje osnov računalniških komunikacij (lastnosti plasti, vzpostavljanje komunikacij, IP protokol).



4.1 KAKO DO VARNOSTI NA SPLETU

Raziskave

Po raziskavi RIS delež uporabnikov svetovnega spleta narašča konstantno in presega 70% vseh anketiranih. 60% jih splet uporablja vsak dan. (Mašić, 2010)

Med dijaki enega razreda prvih letnikov na šoli smo naredili kratko raziskavo. Vsi so zatrjeli, da uporabljajo mobilne telefone za dostop do spleta, nihče pa ni znal odgovoriti, ali dostopajo preko WLAN ali GPRS/UMTS povezave.

Zadnje napovedi največjih nevarnosti v letu 2011 izpostavljajo: sponzorirane napade, napade z uporabo zlonamerne kode v brskalniku, težave z lastništvom datotek na datotečnih strežnikih, varnost podatkov v oblaku, varnostne težave z uporabo mobilnih naprav, povečanje pravnih pritiskov na podjetja, ki ne upoštevajo pravic uporabnikov (<http://www.net-security.org/secworld.php?id=10154>, 6. 5. 2011).



Število uporabnikov spleta narašča, tako pa tudi odgovornost informatikov, da jim zagotovijo varno okolje za delo. Žal je veliko uporabnikov nevedskih oziroma varnosti posveča premalo pozornosti. Vedno več jih uporablja socialna omrežja (<http://www.ris.org/index.php?fl=2&lact=1&bid=11980&parent=27>, 10. 5. 2011), ki spodbujajo uporabnike k skupni rabi datotek in informacij, hkrati pa izvemo, kako avtorji socialnih omrežij kot tudi proizvajalci mobilnih telefonov pridobivajo podatke o uporabnikih njihovih proizvodov in storitev (<http://www.politikis.si/?p=23062>, 25. 4. 2011). Pri varnosti torej ne gre več za programsko in strojno opremo, ampak za vedenje in obnašanje uporabnikov.

Razprava

Razmislite o lastnih navadah pri uporabi računalniških sistemov in storitev. Ali uporabljate antivirus, antispam programsko opremo? Kako pogosto posodabljate sistem, ali so datoteke zaklenjene? Kako imate nastavljene pravice nad mapami, datotekami? Ali poznate procese, ki tečejo na vašem računalniku? Kako imate nastavljen brskalnik? Ali so prijave na uporabljane spletne storitve varne? Ali izdelujete varnostne kopije? Kdaj ste zadnjič zamenjali svoja gesla?

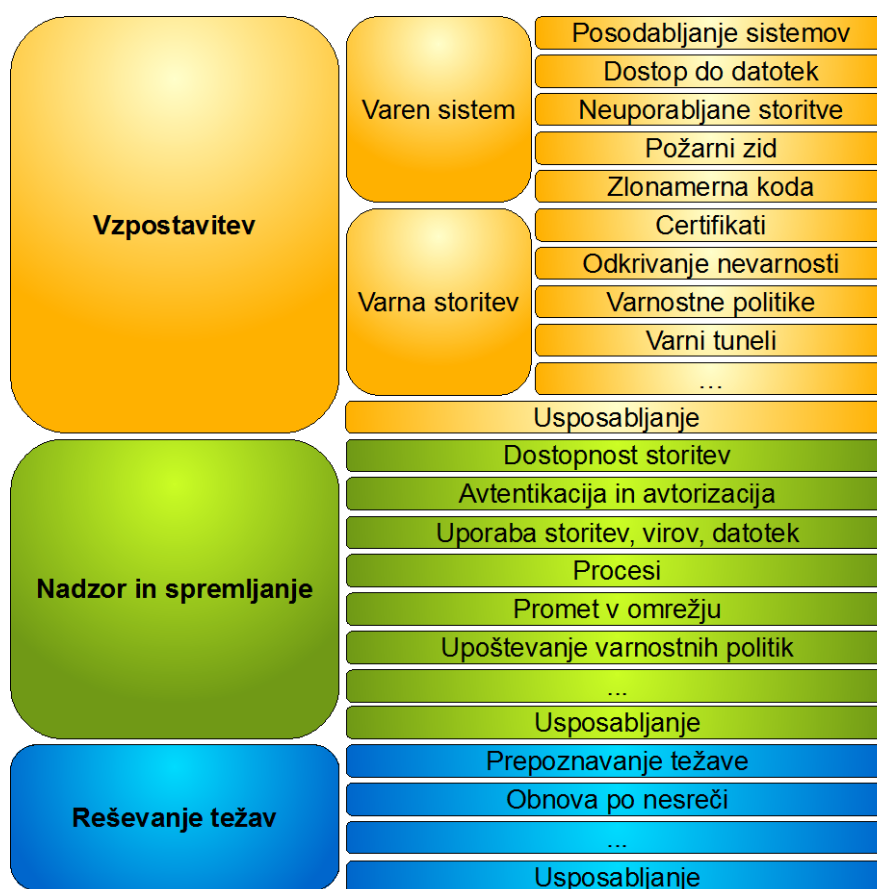


Za podjetja je bistveno, da se s problemom varnosti spopadejo sistematično (Vacca, 2009). Najprej je potrebno ovrednotiti tveganja in grožnje. Pri tem je potrebno biti pazljiv o splošnih prepričanjih, npr. o lojalnosti zaposlenih, znanju informatikov, navadah uporabnikov ipd. Vedno je potrebno poskrbeti za usposabljanje zaposlenih informatikov. Usposabljanje naj bo redna dejavnost tudi za zaposlene, ker je potrebno vzpostaviti kulturo varne rabe storitev. S tem se najprej prične uporabljati vgrajene varnostne funkcionalnosti operacijskih sistemov in aplikacij npr. čiščenje neuporabljenega prostora na USB diskih, izbire dobrih gesel ipd.

Vzpostaviti je potrebno tudi stalen monitoring sistemov, v smislu kdo dostopa do katerega sistema, ob katerem času in kaj s sistemom počne. Sisteme je potrebno redno posodabljanje. Priporočljivo je, da varnost občasno pregleduje nekdo zunanji, ki sistemov ni postavljajal.

4.2 SISTEMSKA PROGRAMSKA OPREMA IN VARNOST

Sistemska opravila v povezavi z varnimi spletnimi storitvami lahko razdelimo v tri kategorije. V prvo sodijo vsa opravila, ki so potrebna za vzpostavitev varnih storitev. Vzpostavitev lahko razdelimo v vzpostavitev varnega računalniškega sistema, ta je osnova za vzpostavljanje varnih storitev. Vzpostavljanje vključuje tudi izvajanje preventivnih ukrepov. Drugo kategorijo sestavljajo opravila spremljanja in nadzora tako komunikacij kot tudi programske opreme, datotek itd. Služba za informatiko mora biti vedno pripravljena tudi na najhujše. V tretji kategoriji najdemo vsa opravila v povezavi z reševanjem težav, od prepoznavanje težave (npr. vdora) do rešitve (npr. obnove po vdoru).



Slika 10: Kategorije opravil

4.3 VZPOSTAVLJANJE VARNIH STORITEV

Osnovni pogoj za vzpostavitev varne storitve je najprej vzpostavitev varnega sistema, ki bo storitev poganjal. V spletu najdemo veliko navodil in priporočil, kdaj je nek sistem varen.

Opisi med drugim vključujejo naslednje ³:

- Redno posodabljanje sistema je eno najpomembnejših vodil. Hkrati je potrebno izpostaviti, da nekritično nameščanje posodobitev včasih privede do slabšega delovanja določenih storitev. Predvsem pri posodabljanju jedra operacijskega sistema lahko izpostavimo Linux, ki omogoča namestitev več jeder, kar omogoča skrbniku sistema enostavno preklapljanje med različnimi verzijami.
- Delo s privilegiranim računom je navada slabih skrbnikov. Za redno delo na sistemu se uporablja nepriviligirani račun. Le ob skrbniških opravilih se preklopi v privilegirani račun. Tu se izpostavlja predvsem namizne verzije Windows sistemov, kjer so največkrat računi uporabnikov privilegirani.
- Uporaba programske opreme za boj proti vsakovrstni zlonamerni kodi (virusi, trojanski konji, spyware) in spletnih straneh (napadi z ribarjenjem).
- Uporaba požarnega zidu omogoča zelo jasno opredeljevanje, katere vrste komunikacij in protokolov bomo na sistemu omogočali in katere ne.
- Opredelitev pravic nad datotekami in mapami. Nekateri sistemi postavljajo pravice privzeto zelo svobodno, kar je v primeru strežnikov nezaželeno. Vsak uporabnik mora imeti dostop samo do tistih datotek in map, ki jih potrebuje za svoje delo. Predvsem osebne mape je včasih dobro šifrirati in s tem onemogočiti dostop nezaželenim.
- Odstranitev storitev, ki se ne uporabljajo. Včasih skrbniki opravijo namestitev sistema s privzetimi namestitvami, ki vključujejo tudi programsko opremo, ki je ne bomo uporabljali. Priporočljivo je, da se programska oprema, ki se ne uporablja, iz sistema umakne.

Raziskovanje in utemeljitev

Uporabite spletni iskalnik za pridobitev informacij o tipih zlonamerne kode: virus, trojanski konj, črv, vohunski program, oglaševalski program, škodljiva programska oprema (*Riskware*). Najdite tudi informacije o goljufijah in prevarah (*scams, hoax*). Naredite razpredelnico z opisi zlonamerne kode in jo dopolnite s podatki o tem, kakšna programska oprema oziroma dejanja vas pred vsako nevarnostjo ščitijo.



Včasih je ob vzpostavitvi varnega sistema potrebno poskrbeti tudi **za varno lokalno omrežje**, v katerem bo storitev potekala. V tem primeru je dobro razmišljati o sistemu nadzora nad omrežjem (*Network Access Control*). Osnovne zmožnosti teh sistemov vključujejo onemogočanje dostopa do omrežja neustreznim računalniškim sistemom, uveljavljanje varnostnih politik glede dostopanja do omrežja, omrežnih naprav in tudi overjanje uporabnikov.

Varnost lokalnega omrežja povečujemo tudi z uporabo požarnih zidov. Ti omogočajo (Egan, 2005):

- filtriranje paketov v smislu preverjanja izvora, cilja in tipa paketa;
- preverjanje stanja in vsebine paketov;
- podpora vzpostavitvi virtualnih zasebnih omrežij.

Povezovanje znanj

Področje upravljanja sistemske programske opreme se v tem delu zelo povezuje s področjem komunikacije. Preverite svoje znanje in naštejte protokole, ki jih različna programska oprema uporablja, in zmožnosti posameznih tipov požarnih zidov.



³ (Smith, 2011)

(http://www.windowsecurity.com/whitepapers/Armoring_Linux.html, 26. 4. 2011)

(<http://www.realonlinemarketing.com/7-ways-to-make-xp-more-secure.html>, 27. 4. 2011)



Slika 12: Arhitektura spletne storitve in povezave med entitetami

Vsaka entiteta predstavlja za drugo entiteto dejansko storitev. Če storitev teče na lokalnem računalniku uporabnika, se označuje kot **lokalna storitev**. Večina storitev je **oddaljenih**, saj se izvajajo na strežnikih, do storitev pa uporabnik dostopa preko spleta. Z rdečo barvo so označene povezave, kjer je potrebno poskrbeti za varno komunikacijo. Povezave med entitetami so lahko:

- lokalne (entitete, ki komunicirajo, so na istem strežniku),
- oddaljene (entitete, ki komunicirajo, so na različnih strežnikih).

Pri vzpostavljanju povezav je potrebno biti pozoren na (Volodarsky, 2005):

- Ali je mogoče do posamezne entitete dovoliti dostop samo z omejenega ali celo samo enega naslova? Če sta npr. aplikacijski strežnik in SUBP na istem strežniku, je smiselno dostop do SUBP omejiti samo na lokalne povezave, podobno pa je včasih smiselno omejevati dostope tudi s strani uporabnika npr. na lokalno omrežje ali celo posamezni računalnik.
- Kdo lahko dostopa do posamezne entitete in kakšne pravice potrebuje (avtorizacija)? Npr. pri nastavljanju dostopa aplikacijskega strežnika do baze podatkov se ustvari račun, ki omogoča dostop samo do izbrane baze z minimalnimi pravicami, ki so potrebne za delo; entiteta tako ne more brisati struktur, ampak ima možnost upravljanja le s podatki.
- Kakšna komunikacija naj se dovoli med entitetami (v primeru oddaljenih povezav)?
- Če je mogoče, naj bo povezava šifrirana (v primeru oddaljenih povezav).

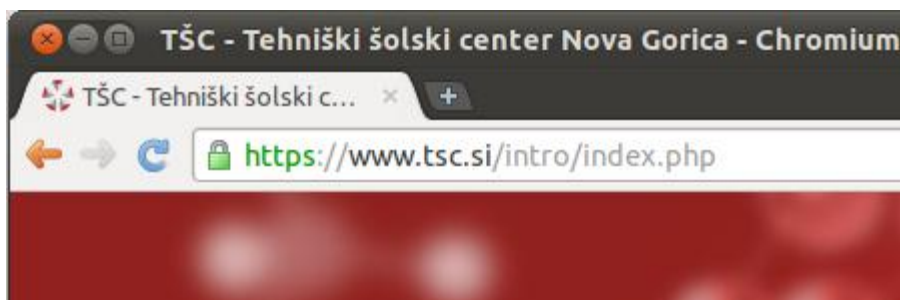
Proces preverjanja identitete uporabnika imenujemo avtentikacija, proces preverjanja pravic uporabnika pa avtorizacija.

Samopreverjanje

Odprite brskalnik in se postavite na začetno stran poljubne pogosto uporabljene spletne aplikacije, ki zahteva prijavo. Vpišite svoje uporabniško ime in geslo in preverite, ali je komunikacija ob prijavi varna. Kakšni načini avtentikacije so varnejši? Na katerih spletnih storitvah se z njimi srečamo?



Na vsaki spletni strani, ki jo obiščete, lahko hitro ugotovite, ali je komunikacija med uporabnikom in spletno storitvijo varna ali ne. Tako naslednja slika prikazuje oznako varne komunikacije ob obisku spletnih strani z enim od priljubljenih brskalnikov. Varne povezavo prepoznamo po oznaki protokola HTTPS, navadni uporabniki pa bodo prej opazili slikico ključavnice. Vse varne storitve danes izkoriščajo šifrirne algoritme. Tako za vzpostavitev varne komunikacije uporabljajo digitalne certifikate, za komunikacijo pa šifrirne algoritme.



Slika 13: Oznaka varne komunikacije v brskalniku

Povezovanje znanj

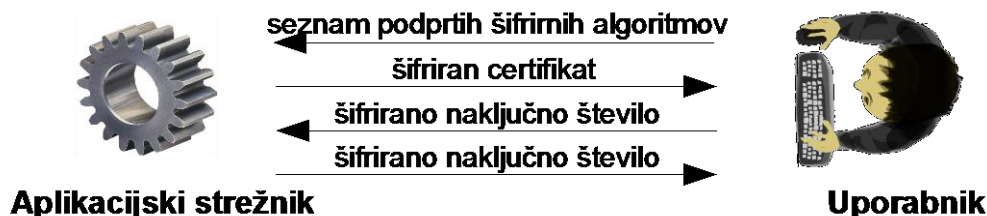
Področje šifriranja se intenzivno obravnava znotraj komunikacij. Za razumevanje delovanja sistemov je potrebno poznati pojme:

- PKI – *Public Key Infrastructure, CA*;
- digitalni certifikat;
- šifrirni algoritem;
- asimetrično in simetrično šifriranje;
- http, https, SSL.



Obravnava naštetih pojmov je izven naših okvirov. Priporočena je uporaba ustreznih spletnih virov in literature.

Za vzpostavitev varne storitve se na strani strežnika postavi sistemska programska oprema, ki za potrebe vzpostavitve varne komunikacije uporablja digitalna potrdila. Postopek ustvarjanja digitalnih potrdil prikazujejo različni online vodiči ([http://pomoc.tsc.si/wiki/Gradiva - računalniški sistemi in omrežja](http://pomoc.tsc.si/wiki/Gradiva_-_raunalniški_sistemi_in_omrežja), 29. 3. 2011).



Slika 14: Proces vzpostavitve varne komunikacije

Slika prikazuje proces vzpostavitve varne komunikacije (<http://onlamp.com/onlamp/2008/03/04/step-by-step-configuring-ssl-under-apache.html>, 27. 3. 2011):

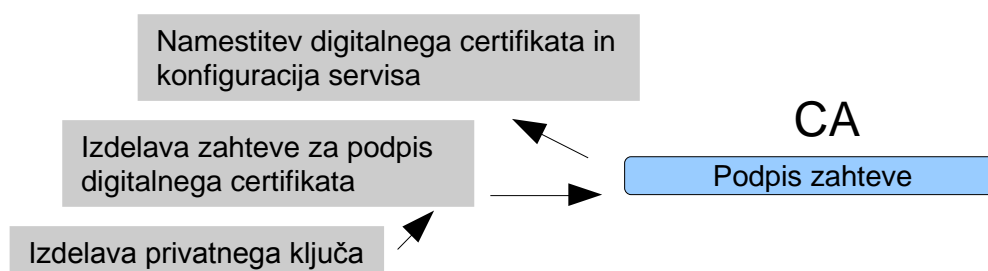
- uporabnik (njegov brskalnik) pošlje strežniku, do katerega želi dostopati, seznam podprtih šifrirnih algoritmov;
- strežnik uporabi najmočnejši algoritem, ki ga poznata oba in odjemalcu pošlje svoj digitalni certifikat z javnim ključem;
- uporabnik preveri certifikat (preko spleta ali z uporabo vgrajenih certifikatov CA) in nato strežniku pošlje naključno število, ki ga šifrira s strežnikovim javnim ključem;
- samo strežnik s svojim zasebnim ključem lahko sporočilo dešifrira in isto naključno število ponovno šifrira, tokrat s svojim zasebnim ključem;
- uporabnik sporočilo dešifrira z javnim ključem strežnika in preveri, če je število enako;
- to število, ki ga poznata samo uporabnik in strežnik, je podlaga za nadaljnjo komunikacijo.

V primerih, ko strežnik zahteva overjanje uporabnika z uporabo digitalnega potrdila, se uporabnikova sporočila šifrirajo tudi z uporabo njegovega zasebnega ključa.

Na strežniku je torej potrebno vzpostaviti ustrezno programsko opremo, ki bo ponujala storitev, programsko opremo za podporo šifriranju in digitalne certifikate.

Koraki so naslednji (podrobni koraki so odvisni od uporabljane programske opreme):

- vzpostavitev storitve brez uporabe digitalnih certifikatov;
- izdelava zasebnega ključa in zahteve za podpisovanje (*Certificate Request*);
- pošiljanje zahtevka agenciji za certificiranje (*Certificate Agency*) (<http://www.sigenc-a.si/>, 6. 6. 2011), za potrebe testiranja in uporabe v zaprtih okoljih lahko certifikat podpišemo tudi sami;
- prejem podpisanega digitalnega certifikata;
- konfiguriranje storitve (aplikacijskega strežnika, poštnega strežnika, ipd.) za uporabo podpisanega digitalnega certifikata.



Slika 15: Koraki pri vzpostavljanju varne storitve

Dostop do varnih storitev javne uprave

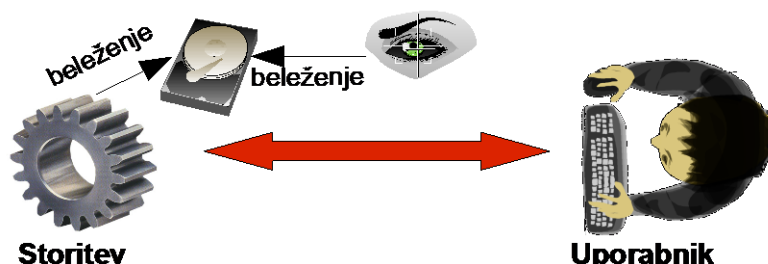
Za dostopanje do varnih storitev javne uprave potrebujete digitalni certifikat, ki ga boste lahko uporabljali tudi za podpisovanje dokumentov in elektronske pošte. Pridobite ga in si ga namestite v spletni brskalnik. A je ta certifikat enak tistemu za spletne strežnike?



4.4 SPREMLJANJE IN NADZOR STORITEV

Osnova za spremljanje in nadzor storitev so varnostne politike podjetja. Varnostne politike so dokumenti, ki opredeljujejo pravilno rabo storitev oziroma opisujejo postopke in vloge v procesu varovanja informacij (<http://www.astec.si/index.php/sl/varnost-in-upravljanje/varnostna-politika>, 27. 3. 2011). Glede na nadzor oziroma beleženje uporabe storitev (beleži se uporabniško ime, tip dogodka, čas, uspešnost dejanja, identiteta spremenjenih podatkov, sistemske komponente ali vira) imamo dve možnosti:

- beleženje dostopov in uporabo storitev opravlja sistemska programska oprema, ki storitev ponuja;
- beleženje dostopov do storitev beleži specializirana programska oprema.



Slika 16: Beleženje na več ravneh

Skrbniki izkoriščajo obe možnosti: nastavljajo storitve, da beležijo podatke o dostopih, in hkrati uvajajo nadzorne sisteme, ki:

- beležijo dostopnost storitev, strežnikov in opreme,
- beležijo obremenjenost posamezne storitve, strežnika, opreme,
- beležijo uporabo protokolov v omrežju,
- beležijo varnost povezav,
- možnost sprožanja programiranih aktivnosti ob dogodkih (npr. ob neodzivanju aplikacije),
- omogočajo opozarjanje ob doseganju mejnih vrednosti,
- izdelajo poročila o vseh dogodkih idr.

Redno spremljanje vseh dejavnosti v podjetju vzame skrbniku veliko časa. Spremljati mora zapise nadzornih strežnikov in zapise vseh storitev, ki jih podjetje ponuja (poštni strežnik, DNS, datotečni strežnik, spletni strežnik ipd.). Vsi sistemi imajo vgrajene zmožnosti opozarjanja skrbnikov ob resnejših dogodkih na elektronski naslov in tudi mobilni telefon.

V podjetjih, kjer je varnost najpomembnejša, pa se skrbniki spopadajo še z:

- testiranjem ranljivosti storitev,
- testiranjem maksimalnih obremenitev storitev,
- centraliziranjem upravljanju vseh popravkov za celotno omrežje idr.

Pomembno je, da je upravljanje informacijskih sistemov zelo dorečeno. Vsak skrbniški račun naj uporablja le ena oseba. Varovani morajo biti tudi zapisi, da jih nepooblaščenim nimajo možnosti spreminjati. Ne nazadnje se morajo skrbniki vedno usposabljanje in vedno izdelovati varnostne kopije podatkov.

4.5 REŠEVANJE VARNOSTNIH TEŽAV

Orodja za vdiranje

Nepridipravi uporabljajo raznolika orodja (<http://www.insecure.in/hacktools.asp>, 28. 3. 2011):

- Brežični vohljači (*Wireless Sniffers*) namestijo brezžična omrežja in pomagajo razbiti zaščito.
- Vohljači paketov (*Packet Sniffers*) analizirajo promet na omrežnem vmesniku, podatki po nezaščitene povezavah so enostavno berljivi.
- Skenerji vrat odkrijejo, katere storitve so na določenem sistemu omogočene.
- Iskalci ranljivosti vrat (*Vulnerability*) skušajo najti varnostne luknje.
- Programe za beleženje vnosov tipkovnice uporabijo za beleženje vsega, kar uporabnik tipka.
- Programe za oddaljeni nadzor.
- Razbijalci gesel iz šifriranega gesla pridobijo nešifrirano.
- Programe za ugotavljanje tipa operacijskega sistema na izbranem sistemu idr.



Nekdo bo z določeno verjetnostjo vdrl v vaš sistem. Ste pripravljeni?

Ne glede ali je vdor malo ali zelo verjeten, mora skrbnik sistemov delovati, kot da se bo vdor kmalu zgodil. Ob tem pristopu je jasno, da ima vsaka organizacija izdelane **načrte za delovanje v primeru vdorov** in v najhujših primerih, **napak** ali celo **katastrofe** (*Disaster Recovery Plan*), ki vključujejo natančna navodila o postopkih in odgovornostih.

V primeru napak ali katastrofe so postopki relativno jasni. Čim prej je potrebno napako odpraviti (zamenjati nedelujoče dele ipd.) in po potrebi iz varnostnih kopij vzpostaviti zadnje konsistentno stanje. Varnostne kopije lahko vsebujejo vse podatke ali pa le podatke, ki so se od zadnjega kopiranja spremenili. V primeru katastrof traja vzpostavitev precej dlje, večja podjetja pa aktivirajo rezervne lokacije.

Težje je dogajanje v primeru vdorov. Vdor je namreč potrebno najprej zaznati. Najpogostejši znaki poskusov vdora so preiskovanje sistemov, številni poskusi povezovanja na sistem iz neznanih naslovov, sprožanje nepopolnih zahtev ipd. Najočitnejši znaki vdora v sistem so:

- uspešno povezovanje na sistem iz neznanega naslova,
- zaganjanje novih storitev na strežnikih,
- spremenjene datoteke na sistemu, predvsem zagonske in konfiguracijske,
- spremenjeni ali zbrisani podatki ipd.

Ko vdor zaznamo, ne moremo zaupati nobenemu sistemu, saj so lahko vsi kompromitirani. Sisteme je potrebno torej umakniti iz delovanja in narediti kopijo vsega. Nato sledi odkrivanje in odpravljanje varnostnih lukenj in postavljanje sistemov v delovanje. Končno sledi rekonstrukcija vdora in odkrivanje krivca.

Povzetek

Pri varnosti sta najpomembnejša vedenje in obnašanje uporabnikov. Pri uporabi storitev Spleta 2.0 so uporabniki aktivni, prispevajo vsebine, komentirajo in ocenjujejo vsebine drugih. Podjetja se morajo s problemom varnosti spopadati sistematično. Sistemska opravila, povezana z varnostjo, razdelimo na vzpostavitev, nadzor in spremljanje ter reševanje težav.

Osnovni pogoj za varne storitve so varni sistemi, ki storitev poganjajo. Varnost omrežij lahko dvignemo z uporabo požarnih zidov. Varnost uporabnika pogojujejo tudi nastavitve brskalnika.

Če naj bo storitev varna, morajo biti varne tudi povezave med vsemi entitetami, ki sestavljajo arhitekturo lokalne ali oddaljenih storitev. Varne storitve danes uporabljajo digitalne certifikate.

V skladu z varnostnimi politikami se opravlja spremljanje in nadzor storitev, ki se vrši v obliki beleženja dogajanja v omrežju in posameznih entitetah storitve. Nadzor se lahko veča s testiranjem ranljivosti in maksimalne obremenljivosti. Če nastopijo težave, je ključen obstoj načrtov za delovanje ob vdorih in nesrečah. Bistveni element obnove stanja predstavljajo varnostne kopije.

Vprašanja in naloge

- 1) Kako naj se s problemom varnosti spopadajo podjetja?
- 2) Naštete in opišite glavna sistemska opravila v povezavi z varnostjo.
- 3) Preučite poljuben računalniški sistem. Ali ga ocenjujete kot varnega?
- 4) Kakšna je varnostna politika za dostopanje do vašega lokalnega omrežja? Sestavite varnostno politiko za dostop do vašega domačega omrežja.
- 5) Ali lahko požarni zid uporabimo za filtriranje vsebin?
- 6) Kakšne so prednosti Spleta 2.0? Katere storitve Spleta 2.0 uporabljate? Katere funkcionalnosti ponujajo?
- 7) Katere funkcionalnosti brskalnikov posegajo v področje varnosti?
- 8) Izberite poljubno spletno storitev in izrišite arhitekturo. Izpostavite komunikacije med

posameznimi entitetami.

- 9) Kakšna je razlika med avtentikacijo in avtorizacijo uporabnika? Pojasnite razliko na primeru spletne aplikacije za urejanje vsebin na spletnem mestu (WCMS).
- 10) Opišite proces vzpostavitve varne komunikacije med uporabnikom in aplikacijskim strežnikom.
- 11) Opišite korake vzpostavitve varne storitve.
- 12) Na zgoraj opisani arhitekturi izbrane storitve izpostavite možnosti beleženja.
- 13) Katere tipe varnostnih kopij poznamo?
- 14) Izdelajte varnostno kopijo vaše osebne mape. Kako obsežna je? Naredite načrt za samodejno varnostno kopiranje mape. Koliko kopij določenega dokumenta boste hranili?

5 VIRTUALIZACIJA

Cilji poglavja in zahtevano predznanje

Poglavje podaja glavne značilnosti virtualizacije. Bralec spozna razliko med fizičnim in virtualnim, predvsem pa spoznava zmožnosti, ki jih tehnologija ponuja. S primeri, s katerimi se srečujemo v podjetjih, spoznamo glavne prednosti, pa tudi slabosti tehnologije in različne implementacije le-te.

Za učinkovito spremljanje in razumevanje vsebin se zahteva poznavanje arhitekture računalniških sistemov, osnovnega delovanja le-teh, uporaba različnih operacijskih sistemov in upravljanja s sistemsko programsko opremo znotraj teh.



5.1 KAJ JE VIRTUALIZACIJA

Virtualizacija je področje, na katerem se v zadnjih letih pojavljajo vedno nove rešitve. Možnosti je toliko, da informatik, ki ne spremlja redno razvoja, hitro izgubi stik in zato sprejema napačne odločitve.

Izziv in razprava (kontinuiteta)

V podjetju DobriInformatik d.o.o. se je oglasil direktor podjetja VarnaBanka d.d. in izpostavil njihov problem. V podjetju imajo strežnik, kjer jim teče programska oprema, ki nadzoruje vse vidike varnosti. Žal je strežnik že zelo zastarel in kaže znake okvar. Direktor je pripravljen kupiti nov strežnik, vendar zahteva, da se proces nadzorovanja varnosti banke nikakor ne prekine.



Razmislite, kako bi v banki vzpostavili nov strežnik in nanj prenesli vso programsko opremo in nastavitve iz starega strežnika brez prekinitve procesa nadzorovanja.

Izziv in razprava (delovanje stare opreme)

V računalniško podjetje RešiVse d.o.o. je poklical direktor podjetja SrednjeMajhno d.o.o. in izpostavil svoje težave. V podjetju imajo strežniško osnovano rešitev, ki upravlja njihovo proizvodnjo. Programska oprema teče v operacijskem sistemu DOS in zaradi programskih posebnosti v druga okolja ni prenosljiva. Strežnik je zaradi vedno večjega obsega podatkov postal premalo zmogljiv. Hkrati si direktor želi postaviti nov strežnik, kjer bi vsi uporabniki centralno hranili vse podatke in na katerega bi bil priključen tiskalnik. Pričakuje rast števila uporabnikov, vendar ne zna oceniti, koliko bo novih.



Postavite se v vlogo podjetja, ki mora ponuditi rešitev. Podajte rešitev glede na zahteve direktorja podjetja SrednjeMajhno d.o.o. Pobrskajte po spletu in pripravite tudi finančno ovrednoteno ponudbo. Utemeljite izbiro komponent ter njihovih zmogljivosti. Upoštevajte naslednje dileme:

- izdelati morate ugodnejšo ponudbo,
- starejši sistemi na novejših strežnikih ne tečejo.

Oba zgornja primera izpostavljata podobne težave, s katerimi se srečujejo podjetja:

- navezanost programske opreme na strojno opremo,
- kvarljivost strojne opreme,
- vezanost upravljanja rešitev na posamezno strojno opremo ipd.

Ključni težavi sta torej minljivost in omejenost računalniških virov (CPU, pomnilnik, V/I naprave). Da bi bili viri dostopnejši ali bolj izkoriščeni, se je že davno začel proces ustvarjanja njihovih virtualnih različic:

- Da bi aplikacije imele več pomnilnika in bi bil hkrati glavni pomnilnik bolj na voljo kodi, ki se dejansko izvaja, je nastal virtualni pomnilnik (*Virtual Memory*).

- Da bi več uporabnikov lahko uporabljalo iste naprave, se je uvedlo virtualne naprave (*Virtual Devices*).
- Končno imamo tudi virtualne računalnike, ki uporabljajo navidezne procesorje, pomnilnik, naprave in navidezno omrežje.

Priporočilo

Wikipedija je odličen vir tudi za raziskovanje različnih vidikov virtualizacije. Začnete lahko na naslovu <http://en.wikipedia.org/wiki/Virtualization> (22. 1. 2011).



Proces ustvarjanja virtualnega iz stvarnega imenujemo **virtualizacija**. Virtualizacija torej ustvari novo abstraktno plast med dvema obstoječima npr. fizičnim virom in uporabnikom le-tega.

Razmislek (abstrakcija)

Kako postopate z vašim mobilnikom, ko kličete prijatelja? Ali iščete v imeniku njegovo ime ali tipkate številko? Ko določeno številko shranimo pod določenim imenom, smo dejansko ustvarili novo plast. Zakaj vse je to dobro?



Virtualizacija z gradnjo abstraktnih plasti zmanjšuje odvisnost procesa upravljanja od posameznih zapletenih elementov npr. vzpostavljanja novih strežnikov, nameščanja programske opreme, hkrati pa nas rešuje težav z minljivostjo virtualiziranih elementov:

- Z virtualizacijo programske opreme se znebimo navezanosti na določen operacijski sistem in tako olajšamo upravljanje programske opreme, ne da bi s tem vplivali na gostiteljski operacijski sistem,
- Z virtualizacijo celotnih računalniških sistemov pa vgradimo konfiguracijo sistemov v strukturo, ki je bolj prenosljiva, lažja za upravljanje in ima več možnosti pri kopiranju in obnavljanju.

Če torej na primer virtualiziramo aplikacije, se ne bomo več ukvarjali s konfiguracijo vsakega sistema posebej. Če virtualiziramo celoten računalniški sistem, ga lahko enostavneje kopiramo in obnavljamo na drugi lokaciji ipd.

Hiter začetek

Večina glavnih ponudnikov rešitev (VmWare, Microsoft, ...) s področja virtualizacije strežnikov ponuja tudi prosto dostopne različice programske opreme za virtualizacijo. Pobrskajte po spletu in hitro lahko najdete več rešitev. Predvsem za preizkušanje in učenje so prosto dostopne rešitve dovolj zmogljive.



Virtualizacija je torej v splošnem pomembna, ker:

- omogoča boljše izkoriščanje strojne opreme,
- zmanjšuje porabo energije,
- olajšuje upravljanje sistemov.

Rešitve znotraj področja virtualizacije lahko kategoriziramo v:

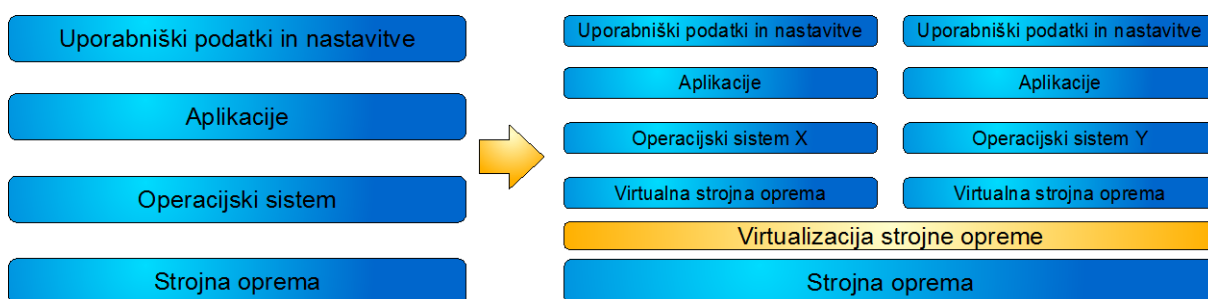
- virtualizacijo strojne opreme (procesorja, pomnilnika, naprav in omrežij),
- virtualizacijo na strani delovnih postaj, kjer govorimo o virtualizaciji namizij, aplikacij in uporabniške izkušnje.

Iz prakse (testna učna okolja)

Podjetje bi rado usposablja svoje zaposlene za rabo najnovejše programske opreme za njihovo področje, ki je dosegljiva na trgu. Kljub temu da je usposabljanje prednost, se podjetje boji padca produktivnosti, če bi zaposlene pošiljalo na usposabljanje drugam. Tako si želi usposabljanja izpeljati v podjetju v laboratoriju. Informatiki v laboratoriju imajo težavo, kako vzpostaviti na istih računalnikih okolja (kombinacijo programske opreme in nastavitev) za različne tipe kadrov. Kako bi tu pomagala virtualizacija?

**5.2 VIRTUALIZACIJA STROJNE OPREME**

Pri virtualizaciji strojne opreme imamo na računalniku nameščeno programsko opremo, katere naloga je priprava simuliranega računalniškega okolja – virtualnega računalnika. V tem okolju se izvaja programska oprema gostujočih sistemov. Sistemsko programsko opremo, ki dodeljuje vire virtualnim računalniškim sistemom, imenujemo hipervizor (*Hypervisor*). Virtualizirane računalniške sisteme imenujemo gosti, osnovni sistem pa gostitelj.



Slika 17: Nova plast – virtualizacija

Gostujoči sistemi se v celoti izvajajo na enak način, kot da bi tekli neposredno na strojni opremi. Hypervisor poskrbi, da dobi vsak sistem ustrezne virtualizirane različice računalniških virov, ki jih potrebuje za svoje izvajanje, in da pri dostopu do fizičnih virov ne prihaja do konfliktov.

Hypervisor omogoča tudi uveljavljanje različnih varnostnih politik za dostop do vhodno/izhodnih naprav, procesorja in ostalih virov.

Spoznavanje zmožnosti orodij za virtualizacijo

S spleta si prenesite poljubno programsko opremo za virtualizacijo strojne opreme. Prostodostopne različice ponujajo vsi večji proizvajalci (VmWare, Microsoft, Oracle, Citrix). Po namestitvi preizkušajte zmožnosti izbranega orodja za virtualizacijo strojne opreme. Kako:

- ustvarimo nove virtualne stroje;
- dodeljemo virtualnim strojem virtualne vire (CPE, pomnilnik, disk, V/I naprave);
- jih zaganjamo, ustavljamo, zamrznemo in zbudimo;
- izdelamo točke povratka in restavriramo stanja;
- nastavimo omrežje?



Za polno razumevanje delovanja je priporočeno daljše preizkušanje različnih rešitev. Nekaj povezav na vodiče dobite na spletni strani <http://pomoc.tsc.si/wiki/VSO>.

Si drznete

- preseliti virtualni računalniški sistem na drug fizični sistem (med delovanjem);
- postaviti strežnike v lastno omrežje?





Slika 18: Prikaz poganjanja enega operacijskega sistema znotraj drugega

V nadaljevanju so podani scenariji uporabe virtualizacije (http://en.wikipedia.org/wiki/Hardware_virtualization, 27. 1. 2011) (Fluks, 2009):

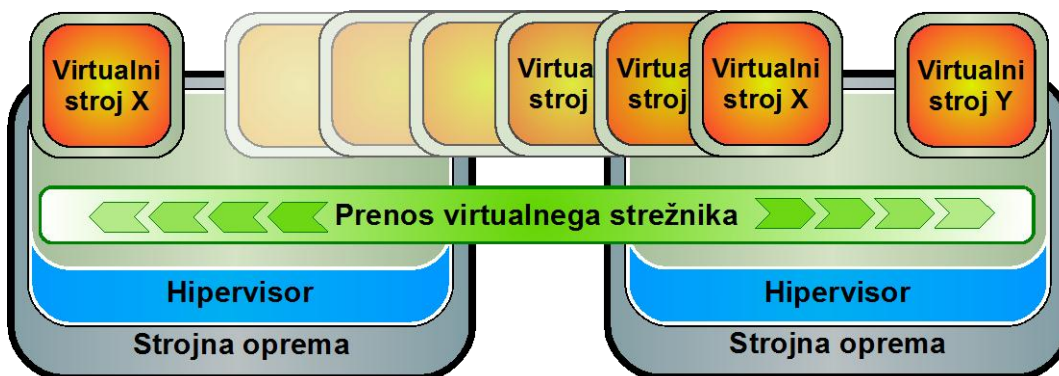
- virtualizacija strežnikov. Da bi strojna oprema bila bolj izkoriščena, se lahko na en fizični strežnik namesti dva ali več virtualnih strežnikov. Tako se prihrani pri strojni opremi, električni energiji, prostoru, vzdrževanju itd.

Raziskava (izkoriščenost opreme)

Raziskave proizvajalcev strojne opreme kažejo, da je povprečen strežnik izkoriščen do 30% (<http://tiny.cc/nqkf7>, 3. 2. 2011).



- preizkušanje in razvoj rešitev za različne operacijske sisteme. Današnja programska oprema mora preizkušeno teči v različnih operacijskih sistemih. Z virtualizacijo je testiranje precej enostavnejše, saj lahko različna okolja vzpostavimo na istem računalniku. Testiranje se s tem najprej poenostavi, saj lahko hitro preklapljam med različnimi operacijskimi sistemi, vzpostavljamo točke povratka, dodajamo in odvezujemo računalniške vire ipd. Ugotovimo lahko, da je testiranje na ta način tudi cenejše.
- Poganjanje nepodprte programske opreme. Veliko uporabnikov uporablja virtualizacijo za poganjanje programske opreme, ki je »gostitelj« ne podpira. Veliko uporabnikov tako uporablja operacijski sistemi Mac OS X in z uporabo virtualizacijskih rešitev poganja aplikacije, napisane za operacijski sistem MS Windows.

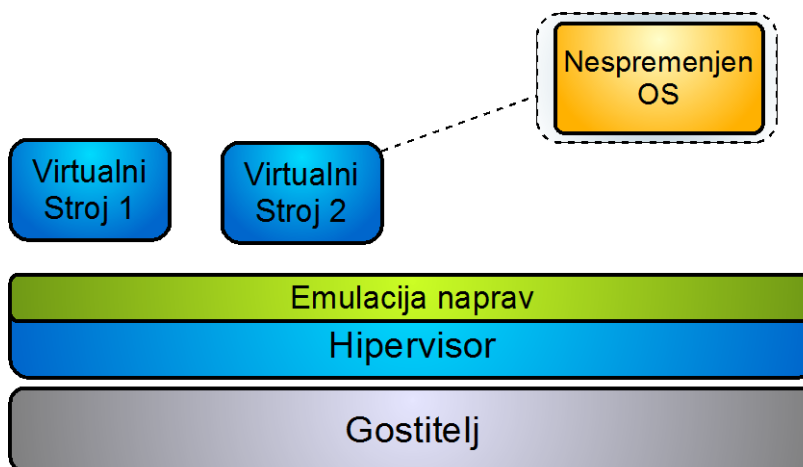


Slika 19: Prikaz premikanja virtualnih strojev med delovanjem

- varnost, zaščita in dostopnost. V večjih podjetjih je varnost in stalna dostopnost storitev nuja. V tem primeru pride do izraza zmožnost prenašanja virtualnega računalnika iz enega fizičnega sistema na drugega. Nekateri sistemi za virtualizacijo omogočajo prenos med samim delovanjem (kor prikazuje slika), pri čemer uporabniki virtualnega sistema prenosa skorajda ne občutijo. Na ta način so aplikacije vedno prisotne, možno je enakomerno obremenjevanje različnih fizičnih sistemov, pa tudi odprava fizičnih napak na sistemih ipd.
- uporaba mobilnih telefonov narašča zelo strmo. Veliko uporabnikov uporablja več telefonov, enega za poslovno in npr. enega za domačo rabo. Z uporabo orodij za virtualizacijo lahko obe rabi združimo v enem računalniškem sistemu. Uporabniki lahko uporabljajo zelo raznoliko programsko opremo, celo drugačen operacijski sistem.

5.2.1 Oblike virtualizacij strojne opreme

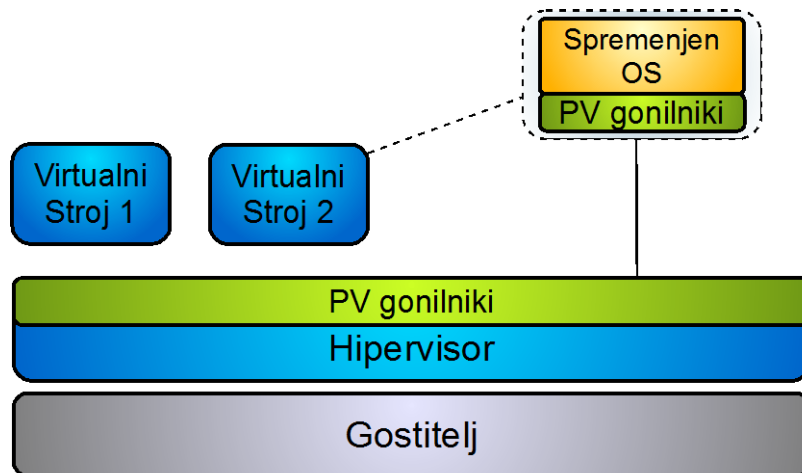
Polna virtualizacija je oblika, ki opravlja emulacijo vse strojne opreme. Rezultat je virtualni stroj, v katerem se lahko izvaja največji nabor obstoječih operacijskih sistemov. Prednost tehnike je popolna izolacija vsakega virtualnega računalniškega sistema. Emulacija vseh naprav pa jemlje vire; tako je polna virtualizacija počasnejša od nekaterih drugih oblik (Desai, 2007).



Slika 20: Polna virtualizacija

Vir: <http://itmanagement.earthweb.com/netsys/article.php/3884091/Virtualization.htm>, (20.1.2011)

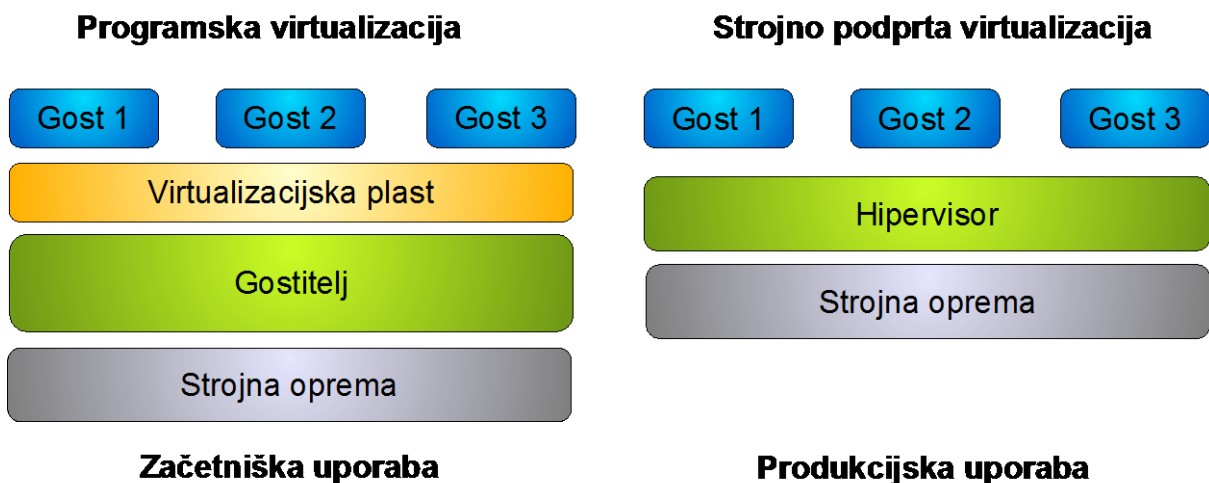
Da bi pospešili izvajanje virtualnih sistemov, se je razvilo več oblik. Ena od rešitev je **paravirtualizacija**. Pri polni virtualizaciji je glavna značilnost in hkrati največji problem polna emulacija naprav s strani hypervisorja, ki terja čas (<http://itmanagement.earthweb.com/netsys/article.php/3884091/Virtualization.htm>, 20. 1. 2011). Pri paravirtualizaciji se problem rešuje na način, da se gostujoči operacijski sistem zaveda, da je virtualiziran. Pri tej obliki se lahko zmanjša kompleksnost komuniciranja med virtualnimi in fizičnimi napravami. Gonilniki gostujočega in gostiteljskega sistema so integrirani (navadno se jim dodaja oznaka PV), kar poenostavlja dodeljevanje fizičnih naprav virtualnim sistemom. Emulacija naprav tako ni potrebna, kar pospeši delovanje gostujočega sistema. Žal paravirtualizacija zaradi navedenega zahteva spremembo jedra gostujočega sistema (<http://en.wikipedia.org/wiki/Paravirtualization>, 25. 1. 2011). Zaradi tega je število podprtih sistemov manjše in v glavnem vključuje odprtokodne operacijske sisteme.



Slika 21: Paravirtualizacija

Vir: <http://itmanagement.earthweb.com/netsys/article.php/3884091/Virtualization.htm>, (20.1.2011)

Zaradi te omejitve so PV gonilnike razvili tudi za polno virtualizirane sisteme. Vendar se v tem primeru zahteva posebna strojna oprema, ki ta način virtualizacije podpira. To obliko virtualizacije imenujemo nativna virtualizacija (*hardware assisted virtualization*) (Desai, 2007).



Začetniška uporaba

Produksijska uporaba

Slika 22: Virtualizacija za učenje/testiranje in produkcijo

Raziskovanje (oznake procesorjev)

Kakšne oznake nosijo procesorji, ki dovoljujejo nativno obliko virtualizacije? Glavna konkurenta AMD in Intel sta za enake rešitve procesorjem dodelila različne oznake.



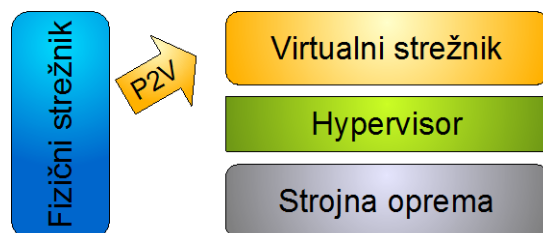
Pri vseh navedenih oblikah imajo gostujoči sistemi lastno jedro. Za razliko od navedenih uporabljajo pri **virtualizaciji operacijskih sistemov** vsi gostujoči sistemi isto jedro. Ker uporabljajo vsi sistemi isto jedro, ta oblika ne podpira izvajanja računalniških sistemov z različnimi operacijskimi sistemi. Virtualni sistemi tudi niso tako izolirani kot pri ostalih oblikah. Velja pa izpostaviti enostavno nameščanje rešitve in hitrost delovanja.

Skupinsko delo (tabela rešitev)

Razdelite se v skupine in sestavite tabelo oblik virtualizacije in hypervisorjev, ki navedeno obliko podpirajo.



Prehod iz uporabe fizičnih sistemov v uporabo virtualnih strojev je včasih zelo težak, včasih pa zelo enostaven. Obstaja tudi več rešitev, ki obstoječi fizični računalnik pretvori v virtualni stroj. Rešitve nosijo oznake P2V (*Physical to virtual*). Proces ponazarja spodnja slika.



Slika 23: Pretvorba fizičnega sistema v virtualni

5.2.2 Virtualizacija diskovnih polj

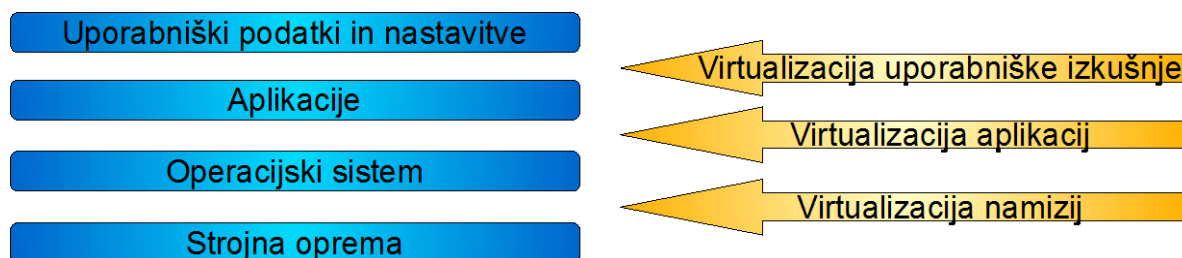
Virtualizacija diskovnega prostora nastopa v več oblikah. V osnovni obliki se več diskov združi in predstavi računalniškemu strežniku in operacijskemu sistemu kot skladna celota. Najbolj znane rešitve so RAID. V sklop te oblike virtualizacije pa štejemo tudi druge programske rešitve, ki več diskov združijo v enega zaradi boljšega izkoristka prostora, zagotavljanja dodatnih funkcionalnosti (varnost, razširljivost) ali hitrejšega delovanja.

S popularizacijo virtualizacije strežnikov se je pojavila povečana potreba po uporabi enotnih fizičnih diskov, ki se po potrebi delijo med več fizičnih strežnikov. Tako se danes uveljavljajo t.i. zunanja diskovna polja, kot so NAS (*Network Attached Storage*) in SAN (*Storage area network*). Predvsem je tu izpostavljen SAN, saj omogoča da več fizičnih strežnikov lahko istočasno dostopa do istih podatkov, kar je ključnega pomena pri migraciji virtualnih računalnikov. Dostop do takega polja je največkrat izveden na dva načina, in sicer s hitrimi omrežnimi povezavami (iSCSI – *Internet Small Computer System Interface*) in posebnimi optičnimi povezavami (FC – *Fibre Channel*). Diskovna polja se operacijskemu sistemu strežnika predstavijo kot enoten disk.

5.3 VIRTUALIZACIJA NA STRANI DELOVNIH POSTAJ

Namen virtualizacije namizij je ločevanje plasti računalniškega namiznega okolja od nižje ležečih plasti. Plasti prikazuje levi del spodnje slike. Največjo težavo predstavljajo močne vezi med posameznimi plastmi. Podatki in nastavitve se zelo navezujejo na uporabljane

aplikacije, te na obstoječi operacijski sistem in ta na strojno opremo. Močne in raznolike naveze povečujejo stroške vzdrževanja. Vsako od teh navez je mogoče pretrgati s pomočjo virtualizacijskih rešitev.



Slika 24: Ravni virtualizacije delovnih postaj

Če ima podjetje na vseh ali vsaj večini računalnikov nameščeno enako programsko opremo, potem bo informatikom blizu razmišljanje o **virtualizaciji uporabniške izkušnje**. S tem želimo doseči nevezanost uporabnika na določen računalniški sistem. Ne glede na to na katerem računalniku v podjetju se uporabnik prijavi, bo imel (Microsoft, 2008):

- dostop do istih podatkov, v primeru uporabe prenosnikov se ti lahko uporabljajo tudi, ko je uporabnik v brezpovezavnem načinu;
- enako namizje in nastavitve;
- enake varnostne nastavitve;
- dostop do istih aplikacij.

Vsi elementi uporabniške izkušnje se hranijo centralizirano, kar olajšuje tudi varnostno kopiranje, in so hkrati dostopni na vseh računalnikih.

Razmislek (nameščanje programske opreme)

Več programov kot na določen sistem namestimo, bolj se delovanje upočasni. A ne bi bilo bolje, če bi vsak program obsegal eno samo datoteko in bi jo lahko enostavno skopirali na računalniški sistem in pognali? Katere so prednosti take rešitve?



Virtualizacija aplikacij opredeljuje programske rešitve (Microsoft, 2008), ki posamezno aplikacijo ogradijo od operacijskega sistema. Programska oprema ustvari plast med njima. Aplikacija zato med delovanjem ne zazna razlike in se obnaša tako, kot bi imela neposreden dostop do operacijskega sistema. Virtualizirana aplikacija ni nameščena na tradicionalen način, se pa izvaja na enak način, kot bi bila.

Izvedb virtualizacije aplikacij je več, za vse pa velja, da je potrebno aplikacijo pred razpečavanjem na računalniške sisteme najprej virtualizirati, kar pomeni izdelati paket, ki se lahko prenaša. Pri rešitvah, ki ne vključujejo strežnika, se potem ti paketi enostavno kopirajo na računalniške sisteme preko omrežja ali prenosljivih medijev in poganjajo. Pri rešitvah, kjer je virtualizacija strežniško podprta, se na odjemalce namesti agente, ki poskrbijo, da se virtualne aplikacije prenesejo na odjemalca na zahtevo in v skladu s pravicami, ki jih določen uporabnik ima. Nekatere od teh rešitev omogočajo prenos samo tistega dela aplikacije, ki jo uporabnik potrebuje (*Application Streaming*) (Microsoft, 2008). Če želi uporabnik virtualno aplikacijo uporabljati tudi v brezpovezavnem načinu, se mora na njegov sistem prenesti celotna aplikacija.

Raziskovanje

V spletnem iskalniku vpišite izraz *free application virtualization*. Med zadetki najdete programsko opremo, ki jo lahko uporabite za virtualizacijo aplikacije. Poskusite virtualizirano aplikacijo pognati na drugih računalniških sistemih.



Prednosti uporabe virtualizacije aplikacij lahko strnemo (Microsoft, 2008):

- aplikacije lahko poganjamo v okoljih, za katere v osnovi niso napisane;
- zaščita sistema pred slabo napisanimi aplikacijami;
- porabi manj računalniških virov kot poganjanje v ločenem virtualnem stroju;
- poganjanje aplikacij, ki med seboj niso združljive;
- poganjanje različnih verzij iste aplikacije;
- ohranjanje enotne konfiguracije računalniških sistemov, ne glede na to katere aplikacije uporabniki uporabljajo;
- poenostavitev prehoda na nove operacijske sisteme;
- hitrejša razpečava aplikacij npr. z uporabo tehnologije application streaming;
- večja varnost operacijskih sistemov zaradi izolacije aplikacij;
- ni potrebno nameščanje aplikacij;
- razpečava aplikacij preko uporabniških profilov.

Virtualizacija namizij ločuje namizje od strojne opreme. Virtualizirano namizje pa se lahko implementira na dva načina:

- na odjemalcu, ko gre za virtualizacijo strojne opreme, kjer v orodju za virtualizacijo poganjamo celoten virtualni stroj;
- na strežniku, ko se celotno procesiranje vrši na njem, na odjemalcu pa se samo prikazuje slika.

Pri virtualizaciji namizij se torej zgodi ograjevanje namiznega sistema in dovoljevanje dostopa do celotnega sistema ali pa prenašanje celotnega namizja na oddaljen računalniški sistem.

Iz prakse (virtualizacija namizij)

Prodajni referent podjetja, ki se ukvarja s proizvodnjo peči, bi rad prikazal programsko opremo za upravljanje peči. Potencialne stranke uporabljajo različne operacijske sisteme in si želijo demonstracij programske opreme za upravljanje peči za uporabljeni operacijski sistem. Kako naj prodajni referent izvrši demonstracijo tako, da bo zadovoljil potrebe potencialnih strank?



Realizacij strežniško podprte virtualizacije namizij je več. V najenostavnejšem primeru gre lahko za upravljanje oddaljenega strežnika (npr. z uporabo VNC, Remote Desktop in Terminal Services v Windows okoljih ipd.). Na ta način lahko do poljubnega namizja dostopamo dejansko s katere koli in kakršne koli naprave. Te rešitve so na voljo že dlje časa. Pravo vrednost pa virtualizacija namizij pokaže, ko razmišljamo, kako bi čim bolj zmanjšali stroške upravljanja z računalniškimi sistemi. Večkrat je ceneje opraviti nakup enega močnega strežnika in veliko t.i. tankih odjemalcev kakor več samostojnih računalniških sistemov. Ker se celotno procesiranje izvaja na strežniku, so lahko odjemalci računalniški sistemi z zelo omejenimi viri, lahko tudi starejši računalniki, ki bi jih sicer v podjetju ne uporabljali več. Take rešitve se označujejo s kratico VDI – *Virtual Desktop Infrastructure*.

Razlika med VDI in oddaljenim upravljanjem je zelo velika. Pri VDI ima vsak odjemalec svoj oddaljeni virtualni stroj, pri oddaljenim upravljanjem pa lahko dela več uporabnikov na istem računalniškem sistemu.

Prednosti (Ruest, 2009):

- enostavno razpečavanje novih verzij namizij,
- zmanjšanje časa nedosegljivosti okolij v primeru okvar strojne opreme,
- zmanjšanje stroškov razpečave (*Deployment*) nove programske opreme,
- daljša življenjska doba strojne opreme odjemalcev,
- varen dostop do namiznih okolij.

Slabosti:

- delna izguba uporabniške avtonomije in zasebnosti,
- izzivi pri upravljanju zunanjih naprav(npr. tiskalnikov),
- težave pri poganjanju zahtevnih aplikacij (multimedija, načrtovanje),
- povečanje nedosegljivosti okolij v primeru okvar omrežja,
- navezanost na omrežje v podjetju,
- kompleksnost upravljanja VDI.

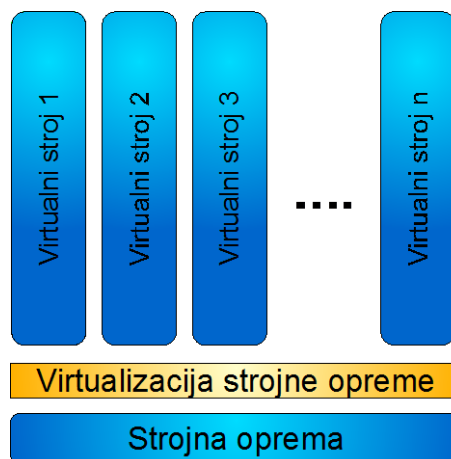
Katero rešitev bi izbrali?

Poskusite se vživeti v naslednje poklice: menedžer podjetja (mobilni uporabnik), raziskovalec (pisarniški nosilec znanja), prodajalka (operativec), zaposlen doma. Napišite, kakšne zahteve ima posamezno delovno mesto in s kakšnimi izzivi se mora spopasti informatik. Za vsako situacijo predlagajte in utemeljite virtualizacijsko rešitev.



5.4 VIRTUALIZACIJA IN RAČUNALNIŠTVO V OBLAKU

Doslej smo se ukvarjali predvsem z rešitvami, ki bolje izkoriščajo en sistem, strežnik ali delovno postajo. S poganjanjem več virtualnih strojev na enem fizičnem smo bolje izkoristili strojno opremo. Strojna oprema in virtualni stroji so torej pri virtualizaciji v razmerju 1:N. Izhajali smo torej iz predpostavke, da so računalniški sistemi premalo izkoriščeni. Vsekakor pa se v praksi srečujemo tudi z drugimi situacijami.



Slika 25: Tipična situacija pri virtualizaciji — ena strojna oprema in več virtualnih strojev

Izziv (hitra rast)

Pri informatiku se je zglasil direktor podjetja HitroRaste in se potožil, da je strežnik, ki so ga nabavili šele pred pol leta, že postal premalo zmogljiv glede na hitro rastoče potrebe podjetja. Informatiku direktor ni znal povedati, kako bodo zahteve podjetja rasle v prihodnje, zaželel pa si je rešitve, ki bi rasla skupaj s podjetjem. Kaj naj informatik svetuje direktorju?

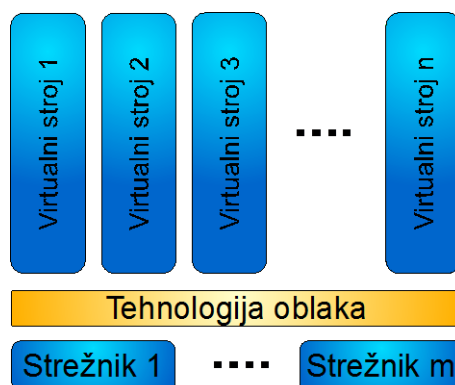


Situacij, kjer je načrtovanje potrebnih virov, je vedno več. Tako se pojavljajo rešitve, ki omogočajo dodajanje virov na zahtevo. Ena od rešitev je tudi **računalništvo v oblaku**.

Opredelitev računalništva v oblaku je veliko, za naše potrebe bomo uporabili opredelitev podjetja Gartner, ki računalništvo v oblaku opredeljuje kot računalništvo, kjer se skalabilni viri dodeljujejo podjetjem kot storitev (SaaS, glej v nadaljevanju). Realizacij tehnologije oblaka je veliko. Osredotočili se bomo na tiste, ki so realizirane s pomočjo virtualizacije. Glavne značilnosti računalništva v oblaku so (Glavač, 2009):

- skalabilnost, elastičnost (kapacitete se lahko prilagajajo glede na potrebe),
- obračunavnje po porabi (plača se toliko, kot se porabi),
- neodvisnost od naprave in lokacije uporabe (do storitve dostopamo preko spleta).

Pri računalništvu v oblaku se torej še bolj zabrišejo meje med fizičnim in virtualnim. Razmerje med virtualnimi stroji in strojno opremo pa se pretvori v N:M, kar je lepo vidimo na spodnji sliki.



Slika 26: Računalništvo v oblaku N virtualnih strojev na M fizičnih strežnikih

Poznamo dva tipa oblakov. Če se podjetje noče ukvarjati z vzdrževanjem strojne opreme, se lahko odloči za najem virtualnega stroja v t.i. **javnem oblaku**. V tem primeru dobi podjetje dostop do svojega virtualnega stroja preko spleta in ga uporabi po želji. Če je podjetje iz kakršnih koli razlogov zadržano do poganjanja virtualnega stroja v spletu, potem bo verjetno vsaj razmišljalo o postavitvi **zasebnega oblaka**. V tem primeru se oblak postavi na strojni opremi, ki je v lasti podjetja. Prvi veliki ponudnik virtualnih strojev v oblaku je podjetje Amazon.

Raziskava

Podjetje Gartner je že 2008 napovedalo, da bo računalništvo v oblaku imelo vsaj tako velik vpliv na podjetja, kot ga ima e-poslovanje. (<http://www.gartner.com/it/page.jsp?id=707508>, 1. 2. 2011)



5.5 OBLIKE STORITEV

Podjetja se morajo vedno obnašati tržno. Tako se velikokrat ne odločajo za nakup opreme, ampak za najem le-te. Pri tem imajo več možnosti (Glavač, 2009):

- IaaS (*Infrastructure as a service*)
Najame se določeno strojno opremo. Ponudnik obračunava obrabo opreme (strežniki, podatkovna skladišča, CPU enote). Danes je vsa infrastruktura, ki se najema, virtualna; tako gre za skalabilno storitev, ki se jo plačuje po porabi.
- PaaS (*Platform as a service*)
Podjetje lahko ob infrastrukturi najame tudi operacijski sistem s pripadajočo programsko opremo. V ceno je zajeto tudi vzdrževanje programske opreme. Poleg operacijskega sistema lahko PaaS zajema tudi drugo programsko opremo npr. za razvoj aplikacij ipd.
- SaaS (*Software as a service*) označuje uporabo programske opreme na zahtevo. Praviloma so SaaS aplikacije spletne aplikacije, ki so optimizirane za uporabo v brskalnikih. Licenciranje je običajno vezano na število uporabnikov, obseg podatkov ipd.

Raziskovanje

V mnogih primerih ustvarjanje novega virtualnega računalniškega sistema, nameščanje operacijskega sistema in dodatne programske opreme sploh niso potrebni. V spletu se za različne hypervisorje dobi že pripravljene pakete (*Virtual appliance*), ki vsebujejo vse naštetu. (http://en.wikipedia.org/wiki/Virtual_appliance, 28. 1. 2011)



5.6 KORAKI PRI UVAJANJU VIRTUALIZACIJE

Uvajanje virtualizacije je proces, ki v podjetje prinaša številne, tudi organizacijske spremembe. Zato je uvajanje virtualizacije potrebno skrbno načrtovati (Ruest, 2009) (Shields, 2008).

Pred uvajanjem kakršnih koli sprememb na področju računalniških sistemov je potrebno najprej zelo dobro **poznati opremo**, ki jo trenutno uporabljamo. Žal veliko podjetjih ne vodi natančne evidence o uporabljeni računalniški strojni in programski opremi. Če evidence ni oziroma je nepopolna, jo je potrebno zgraditi/dopolniti. Pri tem nam lahko pomagajo orodja za raziskovanje lokalnih omrežij. Številna lahko sestavijo seznam:

- uporabljane strojne opreme in programske opreme,
- nastavitve računalniške opreme,
- omrežnih komponent idr.

Raziskovanje

Uporabite spletni iskalnik, kjer vpišite izraz build computer inventory. Med zadetki najdete katero od orodij za raziskovanje lokalnih omrežij. Preizkusite ga na lastnem lokalnem omrežju. Ste presenečeni nad rezultati iskanja? Poskusite najti tudi spletne strani, ki vsebujejo izraz Hardware Assessment Tool, Capacity Planner.



Ko poznamo svojo opremo, moramo ugotoviti, **kaj bi lahko virtualizirali**. Potrebno je najti opremo, ki je kandidat za virtualizacijo. Pri tem ugotavljamo, koliko je izkoriščena strojna oprema, kako se oprema uporablja, ali je kakšna popolnoma neuporabljena ipd. Tega koraka ne moremo opraviti, če ne poznamo rešitev za virtualizacijo. Rezultat koraka je načrt, kaj bo podjetje virtualiziralo.

Redko se zgodi, da obstoječa oprema popolnoma ustreza načrtom za virtualizacijo. Če se npr. odločimo za virtualizacijo strežnikov, bomo verjetno uvedli tudi skupne diske. Če se odločimo za virtualizacijo namizij in aplikacij, bomo mogoče potrebovali dodatno procesorsko moč. Zato opravimo **preverjanje obstoječe opreme** z namenom podpore načrtovanim rešitvam in izdelamo načrt morebitne posodobitve oziroma nadgradnje le-te.

Z virtualizacijo strojne in/ali programske opreme se zgodi tudi arhitekturna zgradba informacijskih sistemov. Nova arhitektura nujno pomeni tudi drugačno upravljanje. Zadnji korak v uvajanju virtualizacije je torej sprememba načina upravljanja informacijskega sistema.

Povzetek

Proces ustvarjanja virtualnega iz stvarnega imenujemo virtualizacija. Virtualizacija sistemov omogoča boljše izkoriščanje strojne opreme, zmanjšuje porabo energije in olajša upravljanje sistemov. Virtualiziramo lahko strojno opremo, namizje, aplikacije ali uporabniško izkušnjo.

Pri virtualizaciji strojne opreme se na računalniku namesti hypervisor. Osnovni računalnik – gostitelj gosti enega ali več virtualnih računalnikov – gostov. Scenariji virtualizacije vključujejo virtualizacijo strežnikov, vzpostavljanje testnih sistemov, poganjanje nepodprte programske opreme, vzpostavljanje varnih okolij idr. Polna virtualizacija opravlja emulacijo vse strojne opreme. Paravirtualizacija zmanjšuje kompleksnost komuniciranja med gostiteljem in gostom, vendar zahteva spremembo jedra operacijskega sistema. Z uvedbo posebne strojne opreme se je razvila nativna virtualizacija. Za razliko od navedenih oblik uporabljajo pri virtualizaciji operacijskih sistemov vsi gosti isto jedro.

Na strani delovnih postaj lahko virtualiziramo uporabniško izkušnjo, aplikacije ali namizja. Pri virtualizaciji uporabniške izkušnje imajo uporabniki enako okolje ne glede na računalnik, na katerem se prijavijo in delajo. V primeru virtualizacije aplikacij se izdelava paketa, ki ga lahko (avtomatizirano) prenesemo na delovno postajo, kjer se vrši v virtualnem okolju. Prednosti rešitve vključujejo boljšo zaščito sistemov in ohranjanje enotne konfiguracije računalniških sistemov, ne glede na to katere aplikacije uporabniki uporabljajo. Virtualizacija namizij se lahko implementira v celoti na odjemalcu ali pa s pomočjo strežnika.

Za maksimalno skalabilnost in neodvisnost od naprave in lokacije uporabe se uporabi računalništvo v oblaku. Podjetje lahko vzpostavi svoj zasebni oblak ali pa uporabi javnega. V primeru najema lahko govorimo o IaaS, PaaS ali SaaS.

Koraki uvajanja virtualizacije vključujejo poznavanje opreme, iskanje kandidatov za virtualizacijo, preverjanja obstoječe opreme in uvajanje novih načinov upravljanja.

Vprašanja in naloge

- 1) Kakšna je razlika med polno in paravirtualizacijo?
- 2) Zakaj potrebujemo za podporo nativni virtualizaciji posebno strojno opremo?
- 3) V virtualni stroj bi radi namestili odprtokodni operacijski sistem Linux. Katero virtualizacijsko obliko bi izbrali in zakaj?
- 4) Pri virtualnih strojih je bistvena prenosljivost. Katero obliko virtualizacije bi uporabili pri ustvarjanju virtualnih strojev?
- 5) Ali res potrebujete zmožnost premikanja virtualnih strojev iz enega fizičnega strežnika na drugega? Zakaj?
- 6) Pojasnite postopek ustvarjanja virtualnega stroja v izbranem okolju za virtualizacijo.
- 7) Kakšne so omejitve pri dodeljevanju virov pri ustvarjanju virtualnih strojev?
- 8) Kakšna je logična zgradba računalniškega sistema? Katere plasti so močno prepletene in se jih splača virtualizirati?
- 9) Ali je mogoče na istem namizju poganjati aplikacije, napisane za različne verzije istega operacijskega sistema? Kaj pa za različne operacijske sisteme?
- 10) Katere težave podjetij rešujemo z uvajanjem virtualizacije?
- 11) Kaj je virtualizacija? Kaj lahko virtualiziramo?
- 12) Naštete scenarije uporabe virtualizacije?
- 13) Kaj je to polna virtualizacija?
- 14) Kaj so prednosti in težave paravirtualizacije?
- 15) Kaj označuje kratica P2V?

- 16) Kakšne možnosti imamo pri virtualizaciji delovnih postaj?
- 17) Ali je mogoče virtualizirane aplikacije dobiti na spletu?
- 18) Zakaj bi aplikacije virtualizirali?
- 19) Kakšna je razlika med virtualizacijo aplikacij in virtualizacijo namizij?
- 20) Kaj označujemo z VDI? Kateri so zadržki pri uporabi?
- 21) V katerih primerih je smiselno razmišljati o uporabi oblaka?
- 22) Kakšna je razlika med zasebnim in javnim oblakom?
- 23) IaaS, PaaS, SaaS – kaj je to?

6 UPRAVLJANJE Z VSEBINAMI

Upravljanje lokalno nameščene programske opreme je lahko zahtevno opravilo. Tako se veliko podjetij odloča za uporabo spletnih storitev. Mednje sodijo tudi sistemi za upravljanje z vsebinami.

Cilji

Poglavje najprej predstavi kategorizacijo sistemov za upravljanje z vsebinami in poda značilnosti le-teh. Bralec nato spozna uporabno vrednost sistemov za upravljanje z vsebinami in spozna povezavo med glavnimi funkcionalnostmi sistemov in standardi.

V nadaljevanju spozna tipične korake pri nameščanju sistemov in upravlja s sistemsko opremo in servisi, ki so potrebni za poganjanje sistemov za upravljanje z vsebinami in strankami.

Bralci z izkušnjami na področju nameščanja in nastavljanja sistemske programske opreme bodo vsebine usvojili veliko hitreje.



6.1 UPRAVLJANJE Z VSEBINAMI

Upravljanje z vsebinami je zaokrožen sklop procesov in tehnologije, ki podpirajo ustvarjanje, zbiranje, upravljanje in objavo informacij v kakršni koli obliki in mediju. V primeru spletnih sistemov za upravljanje z vsebinami je medij svetovni splet. Ena najpomembnejših značilnosti sistemov za upravljanje z vsebinami v spletu (v nadaljevanju sistemov za upravljanje z vsebinami ali tudi CMS – *Content Management Systems*) je enostavnost uporabe, saj je pri uporabniku potreben le spletni brskalnik, s katerim dostopamo na uporabniku prijazen način do vseh zmožnosti CMS-ja. S stališča uporabnika je torej sistem za upravljanje s spletnimi vsebinami storitev, ki je dostopna na nekem spletnem naslovu z uporabo brskalnika. S stališča ponudnika pa je to sklop sistemske programske opreme, ki se jo upravlja na strežniku, ki je dostopen preko spleta.

Primer iz prakse

Pri informatiku se je oglasil lastnik treh turističnih kmetij. Težavo mu predstavljajo rezervacije kapacitet. Dokler je upravljal samo eno kmetijo, ni bilo težav. Sedaj pa izgublja pregled nad zapiski, opomniki, rezervacijami, kontaktni podatki strank ipd. Kljub številnim klicem in povpraševanjem bi bila zasedenost lahko boljša. Lastne spletne predstavitve turističnih kmetij nima. Kaj naj mu informatik svetuje?



Kategorizacij sistemov za upravljanje z vsebinami je veliko. Nekateri viri celo kategorizacijo odpravljajo. S stališča preučevanja v nadaljevanju in lažje aplikacije v praksi izpostavimo naslednja področja:

- Upravljanje spletnih vsebin
Ob uporabi kratice CMS večina uporabnikov pomisli na to kategorijo sistemov, ki vključuje spletne predstavitve, bloge, wikije, forume, upravljanje učnih vsebin, galerije slik ipd. Kategorija se pravilneje označuje s kratico WCMS (*Web Content Management System*).
- Upravljanje z dokumenti
Programska oprema, ki podpira ustvarjanje, samodejno pretvarjanje, nadzorovan dostop in hranjenje dokumentov in zapisov o dokumentih v podjetju.

- Elektronsko poslovanje
Programska oprema, ki neposredno podpira poslovanje podjetja; med drugim vključuje upravljanje odnosov s strankami (CRM – *Customer relationship management*), online nakupovanje, upravljanje z viri (ERP – *Enterprise resource planning*).
- Nadzor izvorne kode
Programska oprema, ki se uporablja pri razvoju programske opreme; osnovna naloga je hranjenje različnih verzij izvorne kode v t.i. vejah (*branch*) razvoja programske opreme. S pomočjo sistemov za nadzor izvorne kode lahko hitro ugotovimo prispevek in aktivnost posameznih razvijalcev itd.

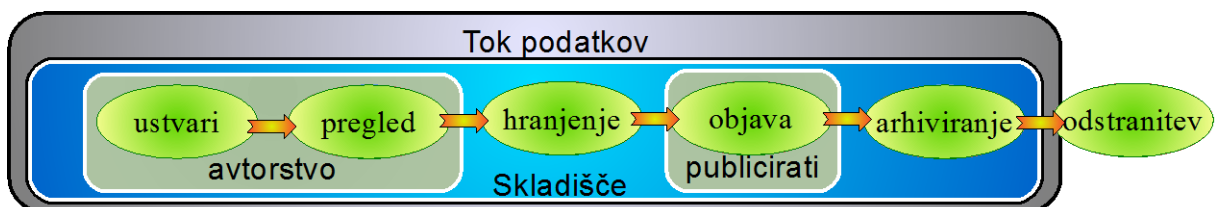
Razmislimo

Z brskalnikom se postavite na spletni naslov <http://www.cmsreview.com>. Preučite zmožnosti sistemov za upravljanje z vsebinami. Simulirajte situacije iz realnega življenja in na tej osnovi določite prednostne zmožnosti CMS-ja. Obiščite še spletna naslova <http://www.cmsmatrix.org> in http://en.wikipedia.org/wiki/List_of_content_management_systems. Ugotovite, v čem se kategorizacije razlikujejo.



Viri navajajo različne kategorizacije zmožnosti CMS-jev. Izpostavimo kategorizacijo, ki zmožnosti CMS-ja razdeli v štiri kategorije (Boyce, 2002), ki so prikazane na sliki, in vključujejo:

- Tok podatkov (*Workflow*)
Je zaporedje korakov, ki opredeljuje pot vsebine od ustvarjanja do objave. Tok podatkov je lahko zelo enostaven, npr. dvodelen ustvari-objavi, ali kompleksnejši kot npr. ustvari-revidiraj-shrani-časovno objavi-umakni. Kljub temu da gre za želeno zmožnost, veliko rešitev ne nudi podpore toku podatkov.
- Shramba
Je okolje, kjer se hranijo vsebine. Željeno je, da se vsebine verzionirajo, kar pomeni, da se lahko po potrebi obnovi starejše verzije. V večini WCMS-jev se vsebine hranijo s pomočjo sistemov za upravljanje baz podatkov. Pri dokumentnih sistemih pa se pogosto uporablja diske za shranjevanje dokumentov in baza za shranjevanje opisov le-teh.
- Avtorstvo je proces, v okviru katerega različni avtorji ustvarjajo vsebine v avtoriziranem okolju. Ustvarjene vsebine lahko obsegajo od nekaj vrstic teksta do multimedijskega gradiva. Ustvarjanje vsebin je lahko sestavljeno iz več korakov. Običajno pisanju sledi revidiranje in končna objava.
- Objava
Je proces, kjer se shranjene vsebine dostavijo na cilj. V primeru WCMS je cilj spletišče, v primeru dokumentov je lahko cilj mapa ali elektronski naslov ipd.



Slika 27: Zmožnosti CMS-jev
Vir: Browing, 2001

Primer

Postavite se v vlogo namišljenega podjetja (majhno družinsko podjetje, podjetje s spletno prodajo, državni zavod, večje proizvodno podjetje ipd.). Simulirajte, kdo v podjetju bi lahko opravljal posamezna opravila znotraj toka podatkov.



6.2 VLOGE V ŽIVLJENJSKEM CIKLUSU CMS-JA

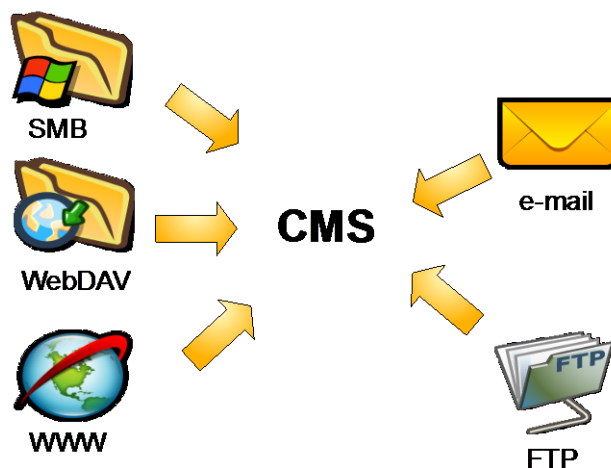
Primer iz prakse

Pri informatiku se oglasi zaskrbljeni direktor majhnega podjetja. Pravkar so vzpostavili spletno stran, sedaj pa ne ve, komu naj zaupa objavo novic. Zadolžena za stike z javnostjo komajda uporablja računalnik. Bo znala napisati novico na spletno stran?



V življenjskem ciklu CMS-ja se srečujemo z različnimi vlogami. Vsaka vloga ima specifična znanja (http://en.wikipedia.org/wiki/Content_management, 18. 2. 2011):

- **Obiskovalec** je vsakdo, kdor z brskalnikom obišče spletišče podjetja. Za to potrebuje osnovna znanja brskanja po spletu in načeloma nima pravic urejanja vsebin.
- **Uporabnik** je vsakdo, kdor v sistemu za upravljanje vsebin slednje tudi ustvarja. Potrebno znanje je odvisno od obsega dela. Za vnašanje novic na portal je potrebno znanje iz uporabe poljubnega urejevalnika. Za delo z dokumentnim sistemom pa poleg znanj s področja klasifikacije dokumentov uporablja tudi orodja, ki do sistema dostopajo preko protokolov FTP, SMB (dostop do dokumentov preko Windows omrežne soseščine) in WebDAV.



Slika 28: Različni načini/protokoli za vnos v vsebin v CMS

V večjih sistemih je uporabnikov več tipov npr.:

- avtor – tisti, ki vsebine ustvarja;
- urednik – tisti, ki vsebine pregleduje, dopolnjuje in razvršča;
- založnik – tisti, ki vsebine objavlja.
- **Skrbnik CMS-ja** mora dobro poznati izbrani CMS. Tipična opravila skrbnika so dodajanje uporabnikov, uporabniška podpora, dodeljevanje pravic, premeščanje in arhiviranje vsebin, kategorizacija vsebin ipd. Najpomembnejši kriteriji sta komunikativnost in doslednost. Med standardi omenimo RSS.
- **Arhitekt** se pojavlja ob vzpostavljanju CMS-ja v podjetju. Običajno gre za zunanega izvajalca, ki ima zelo širok nabor znanj s področja uporabe različnih vrst CMS-jev. Natančno pozna postopke uvajanja, težave ob uvajanjih in podatke o potrebni in dosegljivi strojni in sistemske programske opreme. Obvlada timsko in projektno delo in je odgovoren za uspešno izpeljavo projekta uvajanja CMS-ja.
- V timu ob uvajanju CMS-ja je tudi **oblikovalec**, ki poskrbi za grafično podobo spletišča. Zelo dobro pozna standarde HTML, CSS, orodja in pravila grafičnega oblikovanja. Njegov prvi sodelavec je ob uvajanju WCMS-jev izdelovalec spletnih strani, ki grafično podobo pretvori v ustrezno predlogo za izbrani sistem. Poleg specifičnih znanj iz gradnje predloge za izbrani CMS obvlada standarde HTML, CSS, pozna JavaScript in tehnologijo Flash, Silverlight ipd.

- **Sistemski skrbnik** je strokovnjak za sistemske programske opreme. Pozna vse ključne sestavne dele sistemske programske opreme, potrebne za poganjanje CMS-jev in pripadajoče protokole:
 - aplikacijska plast: PHP, Java, Pyton, Perl, ASP, Ruby;
 - plast baze podatkov: SQL, XML, ODBC, JDBC.

Kaj je trenutno popularno?

V zgornjem razdelku je naštetih veliko standardov. Poiščite opis vsakega posebej na www.wikipedia.org in si izdelajte preglednico ali miselni vzorec uporabljenih standardov. Na spletnih straneh, kjer se opravlja pregled različnih CMS-jev (www.cmsreview.com ali www.cmsmatrix.org), ugotovite, v katerem programskem jeziku je napisanih največ CMS-jev posameznega področja.



6.3 SISTEMI ZA UPRAVLJANJE SPLETNIH VSEBIN

Vedno več uporabnikov in podjetij se želi pojavljati tudi na spletu. Za podporo aktivnostim organizacij na spletu se razvija zelo veliko sistemov za upravljanje spletnih vsebin. Pravilno se označujejo s kratico WCMS (*Web Content Management System*). Ker pa je to področje znotraj sistemov za upravljanje z vsebinami (CMS) zastopano v zelo velikem številu, se uporablja kar kratica CMS. Veliko sistemov je prosto dostopnih.

6.3.1 Uporaba

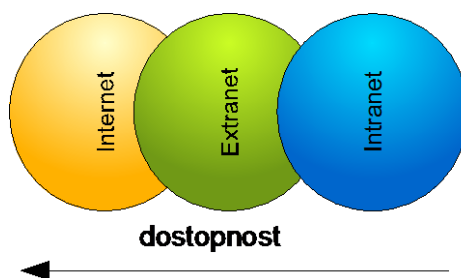
Kaj preizkusiti?

V spletu je dostopnih veliko sistemov za upravljanje spletnih vsebin. Veliko jih je tudi prosto dostopnih. Več jih ponuja tudi demonstracijske namestitve svojih sistemov v preskus. Najdite nekaj takih namestitev (npr. na strani <http://www.opensourcecms.org>) in preizkusite delovanje nekaj sistemov. Zapišite, kakšni so osnovni principi rabe sistemov za upravljanje spletnih vsebin.



Funkcionalnosti sistemov za upravljanje z vsebinami določajo možnosti rabe le-teh. Kategorizacija je bila opravljena v začetnem poglavju. S stališča dostopnosti vsebin in storitev se v podjetjih uporabljajo pojmi:

- intranet opredeljuje storitve in vsebine, ki so dosegljive samo znotraj omrežja podjetja;
- extranet opredeljuje razširitev informacijskega sistema podjetja v smislu dostopnosti od zunaj. Tipično gre dostopanje zunanjih partnerjev podjetja do storitev, ki so dostopna samo zaposlenim. Extranet vedno vključuje določen model preverjanja verodostojnosti, ki ga nadzoruje skrbnik storitev;
- internet ali svetovni splet v tem kontekstu opredeljuje dostopnost storitev vsem, ki lahko dostopajo do svetovnega spleta, splošneje pa označuje mednarodno omrežje omrežij in storitev, ki jih to omrežje omogoča.



Slika 29: Internet, intranet, ekstranet

Vir: Belam, 2006, 19

6.3.2 Prenova spletnega mesta podjetja

Ali je naša spletna stran zanimiva za naše (potencialne) stranke?

Po trditvah podjetja Forrester Research je več kot 60% managerjev podjetij, ki poslujejo v spletu, nezadovoljnih z obstoječimi spletnimi stranmi podjetja.



Spletna stran je danes prva slika vsakršne organizacije. Ker ima velika večina organizacij že tako ali drugačno obliko spletnih strani, ne bomo izpostavljali gradnje, temveč prenovo spletnih strani. Pobude za prenovo dajo velikokrat nezadovoljni obiskovalci. Prva naloga vodstva organizacije je, da te pobude zazna. Nato lahko sledi zagon projekta prenove.

Prvi korak ob prenovi je resna **analiza obstoječega stanja**. Pri obstoječih spletnih straneh je potrebno pozornost posvetiti več vidikom.

Oblikovni vidik prenove spletnih strani

Trendi v oblikovanju se sčasoma spreminjajo. To, kar se je pred leti zdelo kot najnovejši trend, je čez nekaj let zastarelo. Že oblikovna zastarelost je lahko razlog za prenovo spletnih strani. Oblikovanje spletnih strani naj dela oblikovalec in ne računalnikar. Računalnikar – izdelovalec spletnih strani – pa mora oblikovno predlogo postaviti tako, da ni vezana na vsebino. Le tako bo mogoče spletno stran čez nekaj časa ponovno preoblikovati brez bojzani, da bi ob tem morali preurediti še vse vsebine.

Kako si pomagati?

V spletu je za izbrane sisteme za upravljanje z vsebinami na voljo veliko že pripravljenih predlog. Nekatere so plačljive, veliko pa je tudi prosto dostopnih. Uporabite spletni iskalnik (uporabite frazo npr. free web templates in dodajte ime izbranega CMS-ja) in izbrano predlogo postavite v izbrani CMS.



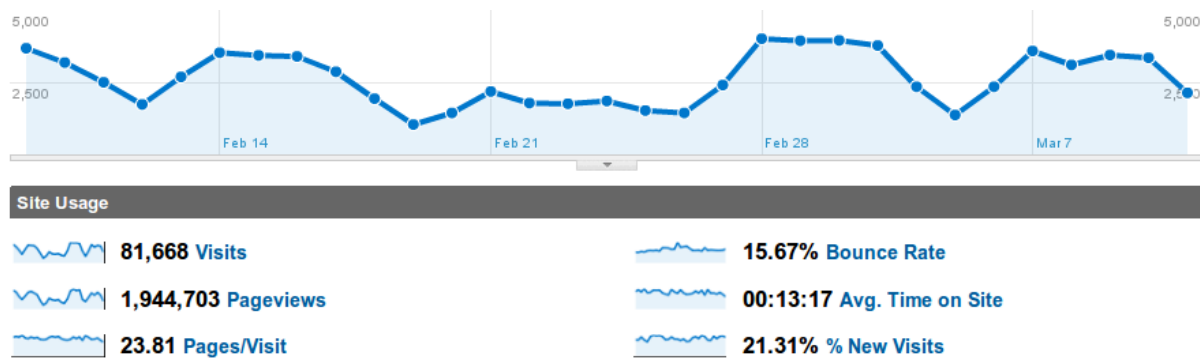
Vsebinski vidik prenove spletnih strani

Vsebinski vidik je obsežnejši. Velikokrat je največja težava **aktualnost vsebin**, ki odraža premajhno zavedanja vodstva o pomenu medija. Če vodstvo ne aktivira zaposlenih in ne določi skupine strokovnjakov, ki vsebinsko skrbi za spletne strani, ne bodo te nikoli aktualne in je vsaka prenova nesmiselna. Sicer se v organizaciji včasih najdejo tudi zanesenjaki, ki sami skrbijo za spletne strani.

Da bo vsebina spletnih strani aktualna, je potrebno v podjetju določiti skupino strokovnjakov. Slednja mora opravljati naslednje vloge:

- Avtorji naj bodo tisti, pri katerih vsebine nastajajo. To ne pomeni, da mora avtor vsebino napisati sam. To velikokrat počne vodstvo. Tako začrta glavne teme vsebine, nadalje pa jo oblikuje.
- Urednik poskrbi, da so vse vsebine zapisane pravilno in v istem slogu. Običajno so uredniki tudi tisti, ki vsebino objavijo, lahko pa imamo za to tudi posebno vlogo.

Pred prenovo je potrebno tudi ugotoviti, katere vsebine uporabnike bolj privlačijo in katere manj. Analiza spletnega mesta se opravi v skladu s ciljno naravnostjo organizacije. Če gre npr. za organizacijo, ki usposablja odrasle, bo analiza vključevala predvsem obisk odraslih. Za analizo obiskanosti se uporabljajo specializirana orodja. Nekatera se namestijo na spletni strežnik in analizirajo podatke, ki jih ustvari, nekatera pa so spletno orientirana in podatke pridobivajo iz skript, ki so vgrajene v spletišče.



Slika 30: Izsek iz analize prometa

Da bi lahko bili še boljši

Za izbrano spletišče opravite analizo prometa. Koliko uporabnikov pride na spletišče s pomočjo iskalnikov, koliko iz drugih strani, kakšno je razmerje med različnimi brskalniki?



Vsebinska analiza vključuje tudi pregled organizacije spletnih strani. Z njim ugotovimo, ali so vsebine smiselno povezane. Ali je globina spletišča na vseh postavkah menija uravnotežena? Velikokrat so meniji nepregledni, preobsežni ali pa vsebujejo preveč ravni. Struktura spletišča mora biti taka, da bodo najpogosteje iskane informacije najlažje dostopne. Za lažjo navigacijo se na strani poleg menijev dodaja tudi oblake oznak, iskalne obrazce in zemljevide strani.

The screenshot shows a website interface with several key elements highlighted by blue callouts:

- zgornji menu**: Points to the top navigation bar containing 'Projekti', 'Izdelki in storitve', 'PUD', 'Zeliščni center', and 'Izobraževanje odraslih'.
- levi menu**: Points to the left sidebar menu with categories like 'O MIC-u', 'Zaposleni', 'Dokumenti', 'Partnerji', and 'Galerija'.
- iskalni obrazec**: Points to the search input field in the 'Iskanje' section.
- oblak oznak**: Points to a tag cloud in the 'Iskanje' section.

The main content area displays search results for 'E-šolstvo', including project listings like 'Predstavitev na prireditvi "Stojnice znanja 2010"'. The right sidebar features sections for 'PROJEKTI', 'NOVICE', and 'POVEZAVE'.

Slika 31: Elementi strani, ki olajšujejo dostop do vsebin

Kritični pogled

Pobrskejte po spletnih straneh in skupinsko sestavite seznam lastnosti dobrih spletnih strani. Primerjajte sezname med skupinami in jih posodobite.



Iz namena uporabe oziroma postavitve spletne strani izhajajo tudi njene funkcionalnosti. Danes se spletne strani ne uporabljajo za enosmerno obveščanje javnosti, temveč se mora s pomočjo spletnih strani ustvariti obojestranska komunikacija. Obiskovalci morajo imeti možnost vplivati na vsebine. Načinov je več (Belam, 2006):

- uporaba kontaktnih obrazcev,
- možnost izpolnjevanja anket,
- možnost komentiranja vsebin,
- možnost pisanja vsebin (članki, forumi, blogi, galerije ...).

Primer iz prakse

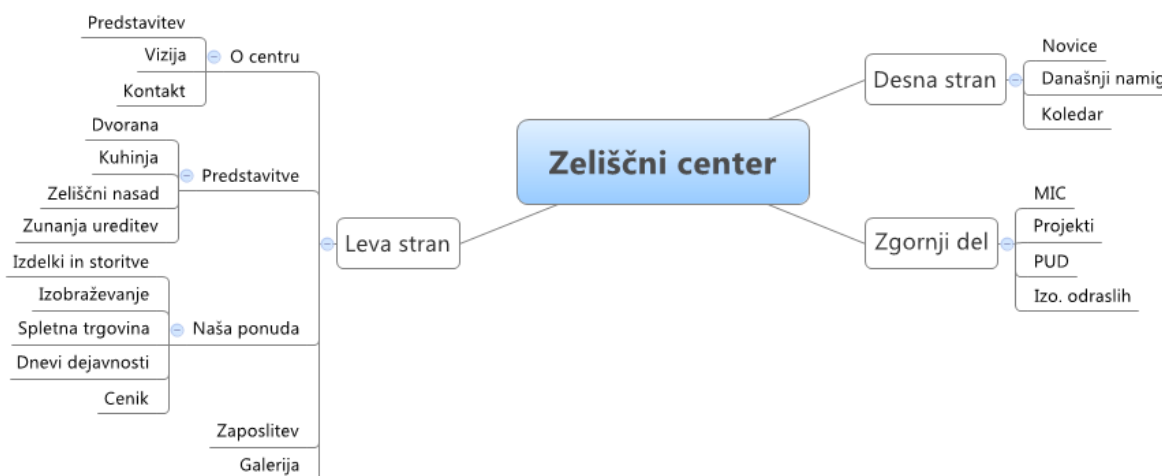
Skušajte razpoznati funkcionalnosti in skicirati spletno stran, če imate naslednji opis: Spletna stran podjetja XYZ d.o.o. mora zagotavljati jasno in atraktivno predstavitev podjetja in dejavnosti, ki jih podjetje opravlja pod lastno spletno domeno www.xyz.si. Omogočati mora različne oblike predstavitve oziroma opisov (lokacija, kontakti, fotogalerija, predstavitev podjetja, opis dejavnosti, videogalerija, novice, ...). S tem podjetju omogoči, da predstavi svoje izdelke oziroma storitve na zelo kakovosten in pregleden način v okviru lastne spletne strani.



V zgornjem delu spletne strani podjetja XYZ d.o.o. naj se izmenjujejo referenčne slike projektov podjetja v določenem časovnem intervalu. V levem delu strani naj bo postavljen meni, ki vsebuje vse elemente predstavitve, povezavo na fotogalerijo in kontaktni podatki podjetja. V desnem delu strani naj bodo postavljene zadnje tri povezave na pretekle referenčne projekte, poleg tega pa naj desna stran vsebuje tudi iskalnik.

Specifikacije za postavitev in organizacijo vsebin lahko podamo v opisni obliki, v ilustrirani obliki npr. s pomočjo miselnega vzorca. Katero obliko se bo uporabilo, je odvisno predvsem od namena in obsega spletnih strani ter stopnje vpletenosti naročnika v projekt izdelave:

- Opisna oblika se lahko uporabi v več fazah projekta izgradnje: lahko le v začetni, kjer opisujemo glavne smernice, ali pa v fazi pisanja natančnih specifikacij.
- Oblika miselnega vzorca je dobra predvsem v fazi zbiranja in usklajevanja usmeritev glede osnovnih zmožnosti in vsebin spletišča.
- Kombinirana oblika vključuje ilustracije in dodane opise. Upoštevamo jih, ko želimo programerjem in oblikovalcem dati zelo jasna navodila o izgledu, razporeditvi in organizaciji vsebin.



Slika 32: Primer vsebinske organizacije spletne strani, podane z miselnim vzorcem

Primer iz prakse – kreativno ustvarjanje

Razmislite, kako bi v izbranem CMS-u izdelali novo sekcijo, kategorijo in objavili članek, spremenili stilsko predlogo v vam najbolj ustrezno, se prijavili v okolje za urejanje strani, spremenili po želji začetno stran, izdelali poljubno anketo, uporabniški meni, vstavili sliko v članek in izdelali uporabniški meni.

**Tehnični vidik prenove spletnih strani**

Ne nazadnje mora stran delovati brezhibno tudi tehnično. Delovati morajo enako v vseh brskalnikih, pri različnih obremenitvah in na različni strojni opremi, tako na strežniški kot odjemalčevi strani.

Bodite pripravljene doseči še več

Z uporabo ustrezne obstoječe programske opreme analizirajte odzivnost izbranih spletnih strani v odvisnosti od števila in aktivnosti obiskovalcev. Najprej izgradite in shranite dejavnost vzorčnega obiskovalca. Nato uporabite zapis dejavnosti in ga multiplicirajte (uporabite lahko programsko opremo BadBoy, Jmeter ipd.).



V nadaljevanju je podan seznam terminov po posameznih fazah življenjskega cikla spletišča:

- zasnova: namen, želeni obiskovalci, zelena vsebina;
- načrtovanje: navigacija, struktura, oblikovanje, zagonski stroški, potrebna znanja;
- izvedba: enostavnost navigacije, enostavnost spreminjanja, stroški, napake, odzivnost, prikaz na različnih napravah, dostopnost;
- vzdrževanje: spreminjanje, dopolnjevanje, aktualnost, stroški.

Uvajanje kakršnega koli sistema v podjetje velikokrat pomeni tudi spremembo načina dela. Že to je lahko velika ovira. Poleg omenjenega pa lahko projekti uvajanja CMS-ja ne uspejo zaradi (Collins, 2010) (Kulla, 2009):

- premalo sredstev,
- premajhne podpore vodstva,
- premalo časa,
- nedorečenih ciljev,
- neupoštevanja pravil gradnje spletnih strani,
- neupoštevanja želja kupcev,
- samogradnje,
- premalo aktivnih vsebin ipd.

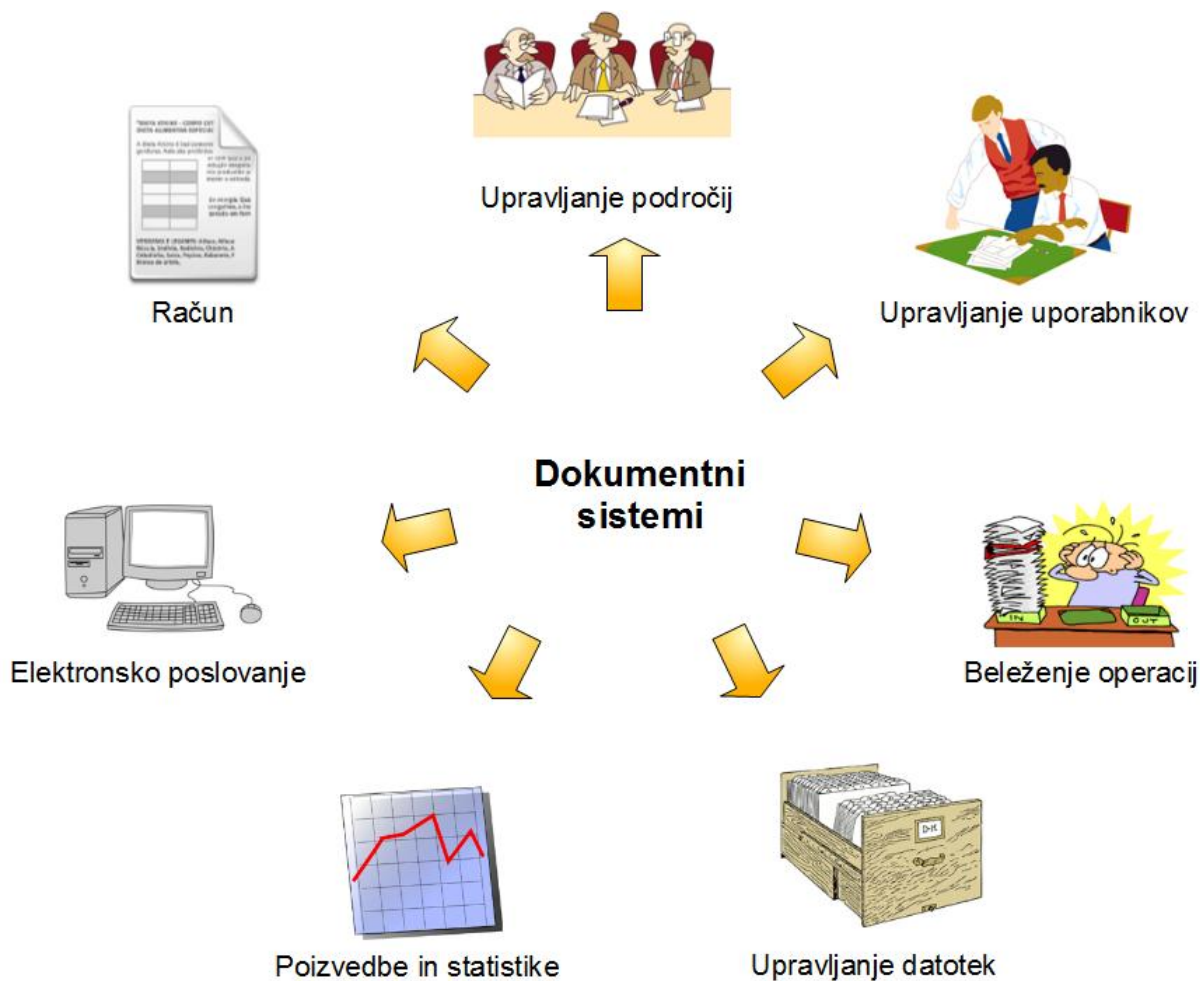
6.4 UPRAVLJANJE Z DOKUMENTI

Elektronski dokument hrani informacije v elektronski obliki, le-te pa so podane v zelo raznolikih oblikah: v tekstovni, slikovni, video, avdio, CAD in drugih oblikah. Kot elektronski dokument lahko smatramo tudi elektronsko pošto. Elektronski dokumenti se hranijo v elektronski in ne fizični obliki. V mnogih podjetjih se dokumenti hranijo po različnih mapah na magnetnih diskih. Na ta način ima v glavnem samo lastnik informacijo o tem, kakšna je vsebina dokumenta in kje se ta nahaja.

Življenjska pot dokumenta v elektronskem dokumentnem sistemu vključuje (Belam, 2006):

- zajem – v sistem lahko pride dokument s preprostim nalaganjem ali prenosom s spleta, lahko pa ga vanj vstavimo tudi preko optičnega čitalnika;
- vpis metapodatkov – metapodatki so lahko prosti, za določene vrste dokumentov pa morajo podjetja upoštevati več zakonov in predpisov (Zakon o elektronskem poslovanju in elektronskem podpisu, Zakon o upravnih postopkih, Zakon o varstvu dokumentarnega in arhivskega gradiva ter arhivih idr.);

- indeksiranje, hranjenje in pretvorba v skladu s pravili;
- posodabljanje, ki ga opravi uporabnik v skladu z dodeljenimi pravicami: uporabniki do dokumentov dostopajo nadzorovano, vsaka sprememba se beleži;
- arhiviranje in varnostno kopiranje in končna odstranitev iz sistema.



Slika 33: Upravljanje z dokumenti

Z uvedbo elektronskega dokumentnega sistema (DMS – *Document Management System*) se v podjetju prične graditi baza znanja o dokumentih, ki se v podjetju hranijo. Lahko izpostavimo naslednje prednosti uporabe elektronskega dokumentnega sistema (Laserfiche, 2000):

- Zmožnost shranjevanja dokumentov in podatkov, neodvisno od izvirne oblike, saj sistem poskrbi za samodejno ali pretvorbo na zahtevo v različne oblike, hkrati pa indeksira celotno vsebino dokumentov.
- Nadzor nad dostopom, distribucijo in spreminjanjem dokumentov. Sistem omogoča natančno določanje pravic nad posameznim ali skupino dokumentov, vsaka sprememba je zabeležena, hrani pa se lahko več ali vse različice istega dokumenta;
- Integracija papirnega in elektronskega dokumentnega sistema. Čeprav podjetje v celoti posluje elektronsko, se vedno najde dokument, ki ga je potrebno hraniti v fizični obliki. Tudi v tem primeru nam elektronski dokumentni sistem pomaga, saj dokument opišemo z meta podatki, ki se hranijo elektronsko.
- Avtomatizacija procesov. Sistemi omogočajo vzpostavitev toka podatkov, kar pomeni, da se dokument po obravnavi pri določenem uporabniku v skladu s pravili takoj posreduje drugemu uporabniku ipd.

Izpostaviti velja tudi zadržke pri uvajanju dokumentnih sistemov. Ob uvajanju le-teh sta vedno potrebni reorganizacija in ponovna opredelitev delovnih mest. Začetek je zelo težak, saj prednosti uvajanja niso vidne, dokler se v elektronski obliki ne upravlja večji del dokumentov. Dejstvo pa je, da se koristi večajo z večanjem organizacije.

Poudariti je potrebno, da dokumentni sistemi niso vsi spletno orientirani. Nekateri imajo razvite lastne odjemalce, ki jih je potrebno namestiti na računalnike vseh, ki bodo dokumentne sisteme uporabljali. Če naj uporaba dokumentnega sistema polno nadomesti uporabo fizičnih dokumentov z elektronskimi, mora dokumentni sistem izpolnjevati vse zahteve zakonodaje in podzakonskih aktov. Načeloma mora izpolnjevanje potrditi ustrezen državni organ.

6.5 SISTEMI ZA UPRAVLJANJE ODNOSOV S STRANKAMI

Namen upravljanja odnosov s strankami je gradnja trajnih in dobrih odnosov med podjetjem in vsemi njenimi strankami. Dobri odnosi so pogoj za trajno sodelovanje in s tem uspešnost podjetja. Gre za strategijo, kjer z uporabo različnih tehnik izvemo čim več o željah naših strank z namenom izdelati boljše ponudbe. Cilji upravljanja odnosov s strankami je ponuditi strankam storitve in izdelke po njihovi meri, ponujanje boljših storitev, pospešiti prodajo storitev in izdelkov, obdržati obstoječe in najti nove stranke itd.

Zavedati se je potrebno moči, ki jo ima stranka

Potrebni je v povprečju 8–10 klicev za prodajo nekega izdelka novi stranki in samo 2–3 klica za prodajo istega izdelka obstoječi stranki.

Razočarana stranka svojo izkušnjo zaupa vsaj 8 ljudem.

Po Paretovem pravilu načeloma 20% strank generira 80% nakupov (Gray, 2001).



Cilje se ne doseže le z nakupom in namestitvijo nekega programa. Uvajanje takega sistema vedno pomeni spreminjanje načinov dela in tudi same organizacije. Koraki uvedbe:

- določiti cilje, želje in izhodišča;
- preučiti možnosti in pripraviti strategijo uvajanja;
- opraviti podrobno analizo, pregledati obstoječe možnosti;
- opraviti izbor med možnimi produkti;
- postaviti in izvesti pilotni projekt;
- usposabljanje uporabnikov;
- testna uporaba in nato produkcija;
- popraviljanje, vzdrževanje v produkciji.

Funkcionalnosti sistemov obsegajo (Kvas, 2010) (Tourniaire, 2003):

- Upravljanje odjemalcev – odjemalci se lahko razporejajo po različnih kriterijih, doda se jim lahko tudi različne informacije (organizacija, delovno mesto, kontakt ipd.), opombe, datoteke (pogodbe, dokumenti, slike itd.), dogodke (klici, obiski, sestanki itd.).
- Delo zaposlenih – vsak zaposleni ureja svoj in skupen nabor odjemalcev: ima svoj urnik in dostopa do skupnih, upravlja svoje in skupne projekte, dogodke, datoteke ipd.
- Različni pogledi na pravice: komercialisti (dostop do operativnih zmožnosti CRM-ja: podpora prodaji, marketingu, klicni center, komunikacija z odjemalci), analitiki (dostop do analitskih zmožnosti sistema: analiziranje, modeliranje podatkov o odjemalcih, prodaji in dogodkih), skrbniki sistema (urejanje pravic za dostop, nastavljanje sistema, urejanje šifrantov ipd.), direktor (poročila, statistike).

Za utrjevanje

V spletu najdete vsaj eno testno namestitev sistema za upravljanje s strankami. Prijavite se vanj z različnimi pravicami in zapišite, kakšne možnosti imajo posamezni tipi uporabnikov.



Opredelitev sistemov za upravljanje s strankami je več, zato so tudi funkcionalnosti različnih izvedb zelo različne. Nekatere vsebujejo tudi funkcionalnosti, ki so del še obsežnejših sistemov. Če CRM-ju dodamo podporo kadrovanju, proizvodnemu procesu, preskrbovalni verigi, projektnemu delu, poslovni inteligenci ipd., dobimo sistem za upravljanje z viri (ERP – *Enterprise Resource Planning*) (Hossain, 2002).

Izziv za študente

Uporabite spletni iskalnik in sestavite seznam sistemov za upravljanje odnosov s strankami. Sestavite preglednico in pri vsakem sistemu zapišite licenco in funkcionalnosti. V katerih podjetjih v domačem okolju bi CRM-je lahko uporabljali?



6.6 UPRAVLJANJE SISTEMOV ZA UPRAVLJANJE Z VSEBINAMI

Ključne faze upravljanja s sistemsko programsko opremo so nameščanje, posodabljanje in varnostno kopiranje. Različni sistemi za upravljanje z vsebinami uporabljajo zelo različne sistemske knjižnice, vseeno pa lahko pri vseh zaznamo enako arhitekturo.



Slika 34: Arhitektura spletnih aplikacij

Sistemi za upravljanje vsebin so spletne aplikacije. Slednje za svoje izvajanje potrebujejo **aplikacijski strežnik**. Ta lahko poleg okolja za izvajanje programov ponuja še druge storitve, kot so upravljanje obremenitev več strežnikov, nadzor nad nenadnimi izpadi, delo v gruči (*Cluster*). Aplikacijski strežnik na eni strani vzdržuje povezave do sistema za upravljanje z bazami podatkov, na drugi strani pa povezave do odjemalcev.

Vse današnje spletne aplikacije delajo s podatki. V večini primerov se ti hranijo v relacijski bazi podatkov, ki jo upravlja **sistem za upravljanje baz podatkov** (SUBP). Do SUBP dostopa aplikacijski strežnik preko ustreznih gonilnikov, večkrat pa aplikacije uporabljajo tudi vmesnike, ki jim omogočajo delo z različnimi SUBP-ji. Med plačljivimi sistemi lahko izpostavimo Oracle, MS SQL Server, med prosto dostopnimi pa MySQL in PostgreSQL.

Aplikacija je napisana v enem od **programskih jezikov**. Vsak zahteva svoje okolje za izvajanje, ki vključuje ustrezen aplikacijski strežnik, včasih pa tudi točno določen operacijski sistem. Med programskimi jeziki danes izstopajo:

- PHP je splošno namenski šifrirni jezik, ki se uporablja za razvoj spletnih aplikacij. Izvaja se lahko na aplikacijskem strežniku z ustreznim predprocesorjem. PHP je dostopen v skladu z licenco GPL. Razvijalci za poganjanje PHP aplikacij večinoma uporabljajo sklop sistemske programske opreme, ki vključuje spletni strežnik Apache, PHP knjižnice in MySQL kot SUBP. Za označevanje naštetih programske opreme se je uveljavila kratica LAMP – Linux Apache MySQL PHP oziroma v primeru Windows operacijskega sistema WAMP.
- Java je programski jezik, ki ga je razvilo podjetje Sun. Podjetje je kasneje večino rešitev objavilo pod licenco GPL. Z Javo lahko razvijamo zelo raznolike aplikacije. V primeru spletnih aplikacij potrebujemo ustrezen aplikacijski strežnik. Obstaja veliko plačljivih (Oracle, IBM), pa tudi prosto dostopnih različic (Apache Tomcat, JBoss) aplikacijskih strežnikov za to platformo. Z uporabo gonilnikov (JDBC) se aplikacije lahko priključijo na katerikoli SUBP.
- C# je programski jezik, ki ga je razvilo podjetje Microsoft in je del ogrodja .NET. Tudi z C# lahko razvijamo zelo raznolike aplikacije. V primeru spletnih aplikacij potrebujemo za izvajanje MS Internet Information Server (MS IIS) in ogrodje .NET. Običajno se podatki hranijo v MS SQL Server-ju, z ustreznimi gonilniki pa se lahko C# programi priključijo tudi na druge SUBP.

Kaj je v ozadju?

V tem poglavju ste že sestavljali sezname različnih sistemov za upravljanje vsebin. Opremite vsak vnos na tem seznamu še s potrebno sistemsko programsko opremo za izvajanje.



6.6.1 Nameščanje

Proces nameščanja sistemov za upravljanje z vsebinami je sestavljen iz več korakov. Le-ti so odvisni od izbranega operacijskega sistema ter sistema za upravljanje z vsebinami in niso vsi vidni.

Prvi korak predstavlja nameščanje ustrezne sistemske programske opreme, ki je pogojen z zahtevami sistema za upravljanje z vsebinami:

- Namestitev SUBP vključuje samo nameščanje programske opreme, nastavljanje le-te in določanje skrbniških pravic.
- Namestitve aplikacijskega strežnika se med seboj zelo razlikujejo: v nekaterih primerih je nameščanje stvar nekaj klikov, v drugih je potrebno kopiranje kode in nastavljanje skript. V primeru uporabe PHP aplikacij je najprej potrebno namestiti spletni strežnik Apache in nato še module za procesiranje PHP skript.
- Podpora povezovanju aplikacijskega strežnika s SUBP se nastavi predvsem z namestitvijo ustreznih gonilnikov.
- Nameščanje dodatnih funkcionalnosti – aplikacije lahko uporabljajo zelo raznolike rešitve (npr. ogrodja za programiranje, različni programski ali vizualni vmesniki itd.). Potrebno je torej poznati zahteve ciljnih aplikacij in namestiti ustrezno sistemsko programsko opremo.

Drugi korak predstavlja nameščanje izbranega sistema za upravljanje z vsebinami. Sistemsko gledano je to spletna aplikacija. Postopek nameščanja lahko strnemo v več korakov (pri nekaterih aplikacijah so lahko določeni koraki avtomatizirani in posledično manj vidni):

- Spletna aplikacija je običajno skupek skript. Ta se prenese s spleta in postavi v mapo, ki jo aplikacijski ali spletni strežnik preslikuje v splet. Običajno se pravice nastavi tako, da ima aplikacijski strežnik možnost zaganjanja skript in po potrebi (odvisno od aplikacije) tudi pisati v določene datoteke oziroma mape.
- Spletna aplikacija vsebuje tudi vsaj eno konfiguracijsko datoteko v XML ali navadni tekstovni obliki. Tu so običajno vpisane razne poti do imenikov na datotečnem sistemu, do spletne aplikacije in nastavitve za dostopanje do baze podatkov (uporabniško ime, geslo in ime baze). Konfiguracijska datoteka lahko vsebuje še druge podatke npr. o glavnem skrbniku, kategorijah vsebin, uporabljenih predlogah, uporabljenih knjižnicah ipd. Konfiguracije se končno lahko hranijo tudi v registru.
- Spletne aplikacije običajno svoje podatke hranijo v bazi podatkov. Nekatere aplikacije znajo delati z bazami v različnih SUBP. Prvi korak je ustvarjanje baze podatkov in uporabniškega računa, ki bo imel pravico upravljanja z bazo, drugi korak pa je ustvarjanje struktur in polnjenje baze z začetnimi podatki, da lahko aplikacija sploh deluje. Te korake se običajno opravi s pomočjo namestitvenih programov.
- Po uspešni namestitvi je spletna aplikacija pripravljena za uporabo, vendar nikakor ne za produkcijo. Sedaj se začne najtežji del, ki je dodajanje vsebin, različnih kategorij, dokumentov, podatkovnih tokov, ustvarjanje uporabnikov in dodeljevanje pravic itd. S stališča uvajanja sistema za upravljanje z vsebinami v organizacijo je ta korak daleč najobsežnejši. Tu je tudi največ možnosti, da delo iz različnih razlogov zastane.
- Po končanih prilagoditvah sledi usposabljanje uporabnikov za delo z novim sistemom in vzpostavitev uporabniške podpore.

6.6.2 Vzdrževanje

Vzdrževanje obsega širok nabor aktivnosti, ki jih opravljajo različni kadri. Večkrat je vzdrževanje vsaj v delu organizirano v obliki zunanjega izvajanja. Vzdrževanje vključuje naslednje aktivnosti:

- Posodabljanje je lahko posledica odkritih nepravilnosti v delovanju skript ali pa odkritih varnostnih groženj. Načini posodabljanja so odvisni od tipov nepravilnosti. V najenostavnejšem primeru gre za prepis skript z novejšimi različicami, pri zahtevnejših posegih pa običajno avtorji napišejo skripte za posodabljanje aplikacije.
- Varnostno kopiranje se opravlja zaradi več razlogov. Najpogostejši je nerodnost uporabnikov, ki prehitro ali po nesreči zbrisejo dokumente. Manj pogost so virusi, črvi, okvare strojne in programske opreme. Varnostno kopiranje mora biti samodejno, saj ročno kopiranje uporabniki žal opravljajo premalo redno.
- Nastavljanje delovanja posameznih elementov programske opreme se običajno opravlja po izvedenih meritvah ali po daljši uporabi sistema, kjer ugotovimo, kolikšno število uporabnikov dostopa do sistema, do katerih delov sistema dostopajo, kako intenzivna je raba ipd. Nastavljanje ima boljšo odzivnost in boljše delovanje sistema.

Povzetek

Upravljanje z vsebinami je zaokrožen sklop procesov in tehnologije, ki podpira ustvarjanje, zbiranje, upravljanje in objavo informacij. Po mnenju uporabnika je sistem za upravljanje s spletnimi vsebinami (CMS) storitev, ki je dostopna na nekem spletnem naslovu z uporabo brskalnika. Po mnenju ponudnika pa je to sklop sistemske programske opreme, ki se jo upravlja na strežniku, ki je dostopen preko spleta. CMS-ji vključujejo upravljanje spletnih vsebin, upravljanje z dokumenti, elektronsko poslovanje, nadzor izvorne kode. Uporabniki CMS-ja so lahko v vlogi obiskovalca, uporabnika, skrbnika, arhitekta, oblikovalca ali sistemskega skrbnika.

Sistemi za upravljanje spletnih vsebin podpirajo trženje podjetij v spletu. Veliko jih je prosto dostopnih. Pri uvajanju se podjetja običajno spopadejo s procesom prenove spletnega mesta. Prenovo sestavljajo faze analize obstoječega stanja, oblikovanje, vsebinska prenova in tehnični vidiki.

Elektronski dokument hrani informacije v elektronski obliki, le-te pa so podane v zelo raznolikih oblikah, v tekstovni, slikovni, video, avdio, CAD in drugih oblikah. Elektronski dokumentni sistemi podpirajo življenjsko pot dokumentov od zajema, vpisa metapodatkov, indeksiranja, hranjenja in pretvorbe do uporabnikovega posodabljanja in končnega arhiviranja oziroma odstranitve iz sistema.

Namen upravljanja odnosov s strankami je gradnja trajnih in dobrih odnosov med podjetjem in vsemi njenimi strankami. Uvajanje sistema za upravljanje s strankami vedno pomeni spreminjanja organizacije. Funkcionalnosti sistemov obsegajo upravljanje odjemalcev, delo zaposlenih in upravljanje pogledov. Sistemi za podporo elektronskemu poslovanju vključujejo tudi podporo kadrovanju, projektному delu, preskrbovalni verigi in proizvodnemu procesu idr.

Ključne faze upravljanja s sistemsko programsko opremo so nameščanje, posodabljanje in varnostno kopiranje. Glavni elementi spletnih storitev so aplikacijski strežnik s podporo izvajanju programskih skript in sistemi za upravljanje baz podatkov.

Vprašanja in naloge

- 1) Kaj označuje kratica CMS?
- 2) Opišite kategorizacijo sistemov za upravljanje z vsebinami.
- 3) Naključno izberite nekaj CMS-jev na spletu in jih kategorizirajte.
- 4) Katere so glavne splošne zmožnosti CMS-jev? Kakšne so zmožnosti poljubnega CMS sistema, ki ga uporabljate?
- 5) Kakšne so vloge pri CMS-jih in kakšna so znanja teh vlog?
- 6) Ali se oblikovalec in sistemski skrbnik morata povezovati? V katerih primerih?
- 7) Kakšna je razlika med intranetom in intranetom? Kaj pa ektranet?
- 8) Kaj vključuje analiza stanja pri prenovi spletnih strani?
- 9) Izberite poljubno spletišče. Kakšne so možnosti uporabnikov za vplivanje na vsebino?
- 10) Kaj zajema življenjska doba dokumenta?
- 11) Kakšne so prednosti uporabe elektronskega dokumentnega sistema?
- 12) Za kaj se uporabljajo sistemi za upravljanje odnosov s strankami?
- 13) Kateri programski jeziki se uporabljajo pri CMS-jih?
- 14) Opišite postopek nameščanja CMS-ja.

7 LITERATURA

Belam, M. *Building an effective web presence in a large organisation*. (online: 1. 3. 2011). Dostopno na naslovu: http://www.currybet.net/cbet_blog/2006/01/gaining-online-advantage-build.php

Bierds, B. et all. *The software deployment mystey – solved: a customer guide*. Redbooks, 2004.

Boyce, J. *Plan and Deploy CMS*. 2002. TechRepublic. (online: 27. 2. 2011). Dostopno na naslovu: <http://www.techrepublic.com/article/plan-and-deploy-content-management-server/5032827> .

Browning, P. in Lowndes, M. *JISC TechWatch Report: Content Management Systems*. University of Bristol. 2001. (online: 22. 2. 2011). Dostopno na naslovu: http://www.jisc.ac.uk/media/documents/techwatch/tsw_01-02.pdf .

Collins, D. *Why Could Your CMS Implementation Fail*. 2010. (online: 24. 2. 2011). Dostopno na naslovu: <http://www.slideshare.net/NavigationArts/why-could-your-cms-implementation-fail> .

Desai, A. *The definitive guide to virtual platform management*. 2007. Realtime publishers. (online: 7. 5. 2011). Dostopno na naslovu: <http://nexus.realtimepublishers.com/dgvpm.php> .

Djurdjič, V. *Odprta Državna uprava*. Monitor. april 2011, stran 23.

Egan, M. in Mather, T. *Varovanje informacij, grožnje, izzivi in rešitve*. Pasadena. Ljubljana. 2005.

Fluks, Z. *Komparativna analiza virtualizacijskih okolij: diplomsko delo*. Fakulteta za Elektrotehniko, Računalništvo in Informatiko. Maribor. 2009. (online: 26. 1. 2011). Dostopno na naslovu: <http://dkum.uni-mb.si/IzpisGradiva.php?id=12540> .

Glavač, Z. *Računalništvo v oblaku in virtualizacija: diplomsko delo*. Fakulteta za Elektrotehniko, Računalništvo in Informatiko. Maribor. 2009. (online: 26. 1. 2011). Dostopno na naslovu: <http://dkum.uni-mb.si/IzpisGradiva.php?id=12781> .

Gray, P. in Byun, J. *Customer Relationship Management*. University of California, 2001.

Hossain, L. Patrick, J.D. in Rashid, M.A. *Enterprise Resource Planning: Global Opportunities & Challenges*. Idea Group Publishing. 2002.

Hriberšek, D. *Horuk v odprto kodo*. Monitor PRO. pomlad 2011, stran 12–13.

Kulla, M. J. *Why do Drupal Projects Fail?: Evaluating Success Factors and When to Use Drupal*. 2009. (online: 24. 2. 2011). Dostopno na naslovu: <http://www.slideshare.net/JuliaKM/why-do-drupal-projects-fail-evaluating-success-factors-and-when-to-use-drupal>

Kvas, K. *Uvedba sistema CRM v podjetju Rauter d.o.o.: diplomsko delo*. Ekonomsko poslovna fakulteta Maribor. 2010. (online: 28. 2. 2011). Dostopno na naslovu: <http://dkum.uni-mb.si/IzpisGradiva.php?id=16685> .

Laserfiche. *Document Management Overview*. 2000. Compulink Management Center Inc.

Limoncelli, T.A. *Time Management for System Administrators*. O'Reilly Media, 2005.

Mašič, S. in Vehovar, V. *Internet in slovenska država*. 2010. Fakulteta za družbene vede – RIS (Raba interneta v Sloveniji) (online: 10. 5. 2011). Dostopno na naslovu: http://www.ris.org/uploadi/editor/1290112352DnD1287670820internet_in_slovenska_drzava_final_21.10.2010.pdf

Microsoft. *Desktop virtualization strategy*. 2008. Microsoft. (online: 8. 2. 2011). Dostopno na naslovu: <http://tiny.cc/ud0zw> .

Moumina, A. *History of Operating Systems*. 2001. (online: 1. 2. 2011). Dostopno na naslovu: http://www.computinghistorymuseum.org/teaching/papers/research/history_of_operating_system_Moumina.pdf .

Ruest, D. in Ruest, N. *Virtualization: A Beginners Guide*. McGraw Hill. 2009.

Ruest, N. in Ruest, D. *Software Deployment Done Right!*. Resolutions Enterprises 2005. (online: 28. 5. 2011). Dostopno na naslovu: <http://www.appdeploy.com/articles/E-book%20Software%20Deployment%20done%20right.pdf> .

Shields, G. *Selecting the Right Virtualization Solution*. Realtime publishers, 2008. (online: 7. 2. 2011), Dosegljivo na naslovu: <http://www.scribd.com/doc/6628310/Selecting-the-Right-Virtualization-Solution> .

Smith, M. *HackerProof: Your Guide to PC Security*. 2011. (online: 26. 4. 2011). Dostopno na naslovu: http://manuals.makeuseof.com.s3.amazonaws.com/MakeUseOf.com_-_HackerProof_PC_Security.pdf .

Tanenbaum, A.S. *Modern Operating Systems, 2nd Edition*. Prentice Hall. 2001. (online: 20. 1. 2011). Delno dostopno na naslovu: <http://www.informit.com/articles/article.aspx?p=24972> .

Tourniaire, F. *Just enough CRM*. Prentice Hall PTR. 2003.

Trampuš, L. *Inšpektor za delo prihaja v vaše podjetje*. Legat. Lesce. 2004.

Vacca, J.R. *Computer and Information Security Handbook*, Morgan Kaufmann Publishers, 2009.

Volodarsky, M. *MSDN Magazine: Design and Deploy Secure Web Apps with ASP.NET 2.0 and IIS 6.0*. 2005. Microsoft. (online: 23. 3. 2011). Dostopno na naslovu: <http://msdn.microsoft.com/en-us/magazine/cc163702.aspx>

Projekt **Impletum**

Uvajanje novih izobraževalnih programov na področju višjega strokovnega izobraževanja v obdobju 2008–11

Konzorcijski partnerji:



Operacijo delno financira Evropska unija iz Evropskega socialnega sklada ter Ministrstvo RS za šolstvo in šport. Operacija se izvaja v okviru Operativnega programa razvoja človeških virov za obdobje 2007–2013, razvojne prioritete Razvoj človeških virov in vseživljenjskega učenja in prednostne usmeritve Izboljšanje kakovosti in učinkovitosti sistemov izobraževanja in usposabljanja.