

Učební texty k státní bakalářské zkoušce
Správa počítačových systémů
Administrace systémů

študenti MFF

15. augusta 2008

6 Administrace systémů

Požadavky

- Instalace systému, plánování síťové topologie, rozklad zátěže
- Zabezpečení, systém práv, správa uživatelských účtů
- Síťové, systémové a adresářové služby, vzdálený přístup
- Zálohování, automatizace úkolů, synchronizace, zotavení systému
- Konkrétní souborové systémy
- Instalace software, hromadná, vzdálená a odložená instalace
- Činnost systému při spouštění a ukončování, konfigurace
- Skriptování a shelly

6.1 Instalace systému, plánování síťové topologie, rozklad zátěže

TODO: tohle je jenom copy&paste z Wiki

Instalace systému

- existuje mnoho variant Unixu a množství distribucí
- distribuce = jádro + balík dalších programů, nástrojů, rozšíření, ...
- FreeBSD, OpenBSD, Solaris, Linux (Debian, Fedora, Gentoo), ...
- software rozdělen do balíčků - volba balíčků k instalaci
- balíčkovací a instalační nástroje - RPM, apt, yum, emerge, ...
- rozdělení disku - /boot, /, /var/log apod.
- volba filesystemu
- konfigurace sítě
- nastavení hesla roota
- po instalaci - kontrola běžících procesů

Plánování síťové topologie

- volba technologií - drátové, bezdrátové
- volba protokolů - dnes zřejmě TCP/IP
- rozvržení lokální sítě, IP adresy (lokální, veřejné), DHCP, klientské stanice
- servery - poštovní, souborové, DNS, zálohovací, webové, aplikační, proxy, ...
- routery, směrování, propojení do Internetu
- sdílení informací o uživatelských účtech (NIS, YP)
- sdílení dat (NFS)
- redundance klíčových serverů a routerů (CARP, master/slave)
- zabezpečení (firewall), šifrování, DMZ (demilitarizovaná zóna), VPN

Rozklad zátěže

- protokol CARP, nástroje CARP a UCARP
- překlad IP adres Round-Robin (NAT)

6.2 Zabezpečení, systém práv, správa uživatelských účtů

TODO: tohle je jenom copy & paste z Wiki

Zabezpečení

- procesy běží v uživatelském režimu s omezenými možnostmi, při systémových voláních se proces přepne do režimu jádra
- preemptivní plánování, priority
- firewall - iptables (Linux), pf (OpenBSD), ipfw (FreeBSD), FW-1 (Solaris), ipfilter aj.
- vypnutí nepotřebných služeb (daemonů)
- zálohování
- sledování logů
- nmap
- Kerberos
- chroot
- sifrovani disku
- sifrovani komunikace
- pouceni uzivatelu
- silna hesla

Systém práv

- každý soubor a proces mají vlastníka a skupinu
- práva pro vlastníka, skupinu a ostatní - čtení, zápis, spouštění
- setuid, setgid - propůjčení práv vlastníka/skupiny při spuštění programu
- setgid pro adresáře - nové soubory budou mít stejnou skupinu jako adresář
- sticky bit pro adresáře - práva k souborům mají jen vlastníci souborů a nikoliv vlastníci adresáře
- uživatel root
- reálné a efektivní UID/GID u běžících procesů
- chmod, chown, chgrp, umask

Správa uživatelských účtů

- /etc/passwd - seznam uživatelů - login, UID, GID, plné jméno, domovský adresář, shell
- /etc/group - skupiny - název, GID, seznam členů
- /etc/shadow - zašifrovaná hesla (hash) - může číst pouze root
- useradd, userdel, usermod, groupadd, groupdel, groupmod, passwd

6.3 Síťové, systémové a adresářové služby, vzdálený přístup

TODO: tohle je jenom copy&paste z Wiki

Síťové služby

- DNS dává jména IP a MAC adresám
- DHCP přiděluje IP adresy
- autentizační služby
- e-mail
- tisk
- NFS (Network File System) umožňuje sdílení souborů a zdrojů

Systémové služby

Adresářové služby

Speciální aplikace pro ukládání záznamů. Typicky se vyhledává hodně a data se mění málo a jednoduše (bez transakcí). Používá se pro uložení údajů o lidech a zdrojích (tiskárny).

Pro přístup se používá standard LDAP, který byl navržen jako odlehčená verze protokolu X.500 ze světa ISO/OSI.

Implementace adresářových služeb

- NIS od Sun-u (příkazy začínají na yp kvůli starému názvu Yellow Pages)
- Active Directory od MS
- OpenLDAP, Kerberos (open-source)

Vzdálený přístup

- SSH - Secure SHell
- Telnet

6.4 Zálohování, automatizace úkolů, synchronizace, zotavení systému

TODO: tohle je jenom copy & paste z Wiki

Zálohování

- nástroje dump, restore, rdump - možno inkrementální režim
- tar + bzip2/gzip
- dd
- Amanda

Automatizace úkolů

- cron
- at

Synchronizace

- rsync - soubory
- NIS - uživatelé

Zotavení systému

- fsck

6.5 Konkrétní souborové systémy

ext2 (second extended filesystem)

- Prostor je rozdělen na bloky a ty jsou uspořádány do skupin bloků. To je kvůli snížení vnitřní fragmentace a minimalizaci pohyb hlav na disku.
- Každá skupina bloků obsahuje: superblock, mapu skupiny bloků, mapu inodů a bloky dat.
- **superblock** – obsahuje základní (nezbytné) informace o svazku. Jeho kopie je tedy v každé skupině bloků
- **i-node** – obsahuje informace o souboru
 - počet odkazů na soubor
 - vlastník, skupina
 - přístupová práva
 - typ souboru
 - velikost souboru
 - časy – modifikace, přístup. . .
 - odkazy na datové bloky
 - neobsahuje název souboru, ten je uložen v adresáři kam soubor patří
- velikost FS daná při formátování

ext3

- nadstavba na ext2 a tedy možnost přechodu mezi těmito dvěma FS bez nutnosti přenášení dat.
- hlavní novinka je, že přibyl žurnál. Jsou možné tři úrovně žurnálování:
 - **Journal** – (nejpomalejší, ale nejspolehlivější) metadata i obsah souborů jsou zapisovány do žurnálu, ještě před komitnutím do FS.
 - **Ordered** – (středně rychlý, středně riskantní)
 - **Writeback** – (nejrychlejší, ale riskantní - v některých ohledech jako ext2) žurnálována jsou pouze metadata, ne již obsah souborů.
- při havárii systému okamžitá oprava podle žurnálu, FS se nemusí kontrolovat

- FS roste za běhu jak je potřeba

ReiserFS

- žunálování metadat
- růst velikosti FS za běhu, dle potřeby
- „tail packing“ pro snížení míry fragmentace
- metadata, adresářové položky, inody a konce (tail) souborů jsou uloženy v jednom *B+* stromě podle univerzálního ID
- mnohdy rychlejší než ext2 (ext3), kde jsou adresáře jako seznamy souborů a tak projití adresáře je lineární na rozdíl od logaritmického u stromu v ReiserFS.

FAT32 (File Allocatio Table)

- jednoduchá implementace \Rightarrow téměř na každém OS
- žádná ochrana proti fragmentaci
- zjistit kolik je volného místa znamená projít lineárně celý disk
- rozdělení disku:
 - Boot sector
 - FAT – dvě kopie FAT (mapy datové oblasti)
 - Datová oblast – soubory a adresáře až do konce disku

6.6 Instalace software, hromadná, vzdálená a odložená instalace

TODO: dodělat, předělat... tohle je jenom copy & paste z Wiki

Instalace software

- software distribuován v balíčcích
- balíčkovací a instalační nástroje - RPM, apt, yum, emerge, ...
- většina software také distribuována jako zdrojový kód
 - stáhnout zdrojové kódy, rozbalit
 - ./configure
 - make
 - make install

6.7 Činnost systému při spouštění a ukončování, konfigurace

TODO: tohle je jenom copy & paste z Wiki

Spouštění systému

- start zavaděče (LILO, GRUB)

- nahrání kernelu
- spuštění kernelu, detekce HW, spuštění ovladačů
- mount root readonly
- spuštění procesu init
- kontrola disků
- re-mount root read-write
- start-up skript (/etc/init.d)
- běžící systém, spuštění konzolí (getty)
- při spouštění možnost aktivovat single-user režim
- úrovně běhu (SystemV, Linux):
 - 0 - systém zastaven
 - 1 - single-user
 - 2 - multi-user, bez sítě a bez NFS
 - 3 - multi-user
 - 5 - multi-user + X11
 - 6 - reboot
 - konfigurace v /etc/inittab

Vypnutí systému

- ukončení programů, zastavení služeb, signál TERM, po chvíli KILL
- vyprázdnění diskových cache, uložení dat, odpojení disků
- vypnutí napájení
- shutdown - možno nastavit čas vypnutí a zaslat oznámení všem přihlášeným

Konfigurace

- na Unixových systémech se konfigurace ve většině případů provádí editací textových souborů
- většina konfigurace schována v adresáři /etc

6.8 Skriptování a shelly

Uplně přesně nevím co by tady mělo být, tak aspoň takovej přehled.

Skriptování

Skriptovací jazyk je programovací jazyk, který je interpretován přímo jak je zadáván z klávesnice. Nedochozí tedy k tomu, jako u klasických programovacích jazyků, že by byl program nejdříve přeložen do binární podoby a potom spuštěn. Skript zůstává ve své původní podobě a je vyhodnocován příkaz po příkazu, tak jak je zadáván. Skriptovací jazyky byly vytvořeny pro urychlení standardního vývojového cyklu editace – kompilace – linkování – spuštění a také pro možnost automatizovat některé úlohy.

Skripty mohou být také zkompileovány, ale protože napsat interpret je jednodušší než napsat kompilátor, tak jsou mnohem častěji interpretovány. Skriptovacích jazyků je obrovské množství.

Skriptovací jazyky aplikací – Velká většina rozsáhlých aplikací obsahuje svůj vlastní skriptovací jazyk, ušitý přesně na míru požadavkům konkrétní aplikace. Například některé počítačové hry používají skripty pro popis chování postav, které nehraje člověk.

- Emacs Lisp,
- Matlab,
- QuakeC,
- UnrealScript,
- Vim scripting language.

Skriptovací jazyky pro WEB – Důležitá součást rodiny skriptovacích jazyků jsou jazyky používané k tvorbě interaktivních webových aplikací.

- ASP,
- PHP,
- JavaScript,
- VBScript.

Jazyky pro zpracování textu – Jedno z nejstarších použití skriptovacích jazyků bylo automatické zpracování textových dat. Spoustu jich bylo původně navrženo jako pomoc administrátorům při zpracování textových konfiguračních souborů a později až dorostly do právoplatných skriptovacích jazyků.

- awk,
- Sed,
- XSLT.

Obecné skriptovací jazyky – Některé jazyky jsou přímo určeny pro nejširší použití a nejsou vázány na nějaké konkrétní použití.

- Lisp,
- Perl,
- Python,
- Ruby,
- Tcl.

Job control languages – Hlavní skupina skriptovacích jazyků vznikla za účelem spouštění a kontrolování běhu programů. Většinou bývají navázány na nějaký operační systém, ale mohou fungovat i na různých architekturách.

- AppleScript,
- bash, csh, ksh . . . ,
- cmd.exe, command.com.

Shelly

Unix shell je tradiční uživatelský interface pro operační systém Unix, nebo systémy na unixu založené. Uživatelé řídí práci počítače přímo psaním textových příkazů pro shell. Pro OS Windows existuje obdoba zvaná *command.com*, nebo *cmd.exe*.

V nejobecnějším významu termín *shell* znamená jakýkoliv program, který uživatel používá k zadávání příkazů. V OS Unix si uživatel může vybrat jaký shell bude používat, proto jich bylo vyvinuto nepřeberné množství. Název shell (skořápka, ulita, plášť . . .) proto, že "schovává" detaily pod ním ležícího operačního systému za svůj interface.

Výraz shell také znamená nějaký konkrétní program, jako třeba *Bourne shell*, nebo *Korn shell*. Bourne shell byl použit u prvopočátků operačního systému Unix a stal se de facto standardem mezi shelly. Každý Unixový systém má alespoň jeden shell s ním kompatibilní. Program Bourne shell je v Unixové hierarchii uložen v `/bin/sh`. Na některých systémech, jako třeba BSD, je `/bin/sh` přímo Bourne shell, nebo jeho ekvivalent, na linuxu je to většinou link na kompatibilní, ale rozšířený a mnohem mocnější shell.

Bourne shell (sh) – původní Unixový shell, který napsal Steve Bourne v Bell Labs. Chybí mu některé věci pro interaktivní práci (doplňování příkazů, manipulace s historií, editace příkazové řádky . . .), ale již obsahuje jednoduše použitelný jazyk pro psaní shell skriptů. Dnes se používají spíše modernější shelly pro svou větší uživatelskou přátelskost.

C shell (csh) – shell používající syntaxi podobnou jazyku C. Je o trochu šikovnější pro interaktivní používání (přidává aliasy a příkazovou historii), ale zase o něco nešikovnější pro psaní skriptů.

TC shell (tcsh) – csh obohacený o doplňování příkazů, editaci příkazové řádky a další vylepšení.

Bourne again shell (bash) – shell nově napsaný ve Free Software Foundation v rámci GNU iniciativy. Obsahuje jazyk pro psaní skriptů použitý v *sh*, ale přidává spoustu užitečných funkcí pro interaktivní používání.

Korn shell (ksh) – rozšíření *sh* a tedy s ním zpětně kompatibilní. Velice mocný nástroj i pro interaktivní používání i pro skriptování. Hlavní výhoda proti ostatním shellům je propracovanost jeho použití jako programovacího jazyka.

Z shell (zsh) – moderní shell vzniklý rozšířením *sh* a přidáním velkého počtu vylepšení a užitečných věcí z *bash*, *ksh*, *tcsh*. Obsahuje například: programovatelné doplňování příkazů, sdílení historie příkazů mezi všemi běžícími shelly, kontrola syntaxe, široce nastavitelné možnosti pro vzhled a chování promptu. . .

Základní shellové nástroje

- cat, grep, head, tail, wc, tee

- cp, rm, mv
- ls
- cd, pwd, mkdir, rmdir
- echo
- more, less
- read
- sort, cut, tr
- find
- xargs
- sed - stream editor
- awk
- a desítky dalších ...

Skriptování v shellu

- roury (pipe)
- standardní vstup, standardní výstup, chybový výstup
- proměnné, speciální proměnné \$X, uvozování, parametry skriptu
- proměnné prostředí
- subshelly
- podmínky, cykly
- funkce
- přesměrování – |, &, »
- signály
- expanzní znaky, regulární výrazy