Tru64:
Uvod - alati i naredbe

Dinko Korunić,
InfoMAR d.o.o.
v1.2, travanj 2006.

O predavaču

- višegodišnji vanjski suradnik časopisa Mrež@, vlastita kolumna "Digitalna radionica - Linux", itd.
- vanjski suradnik SRCE-a: forenzike provaljenih sustava, izgradnja sistemskih paketa, helpdesk za sistemce, sigurnost Unix baziranih sustava, predavač, itd.
- sigurnosni ekspert pri InfoMAR d.o.o.

Tijekom prezentacije

- **ako što nije jasno - pitajte i tražite objašnjenje!**
- ako što nije točno - **ispravite**
- **diskusija** je poželjna i produktivna
- ako je prebrzo - tražite da se **uspori**
- **podijelimo** zajedno vlastita iskustva
- kad vam dojadi - zatražite **pauzu**

Dio I:
Uvod u Unix

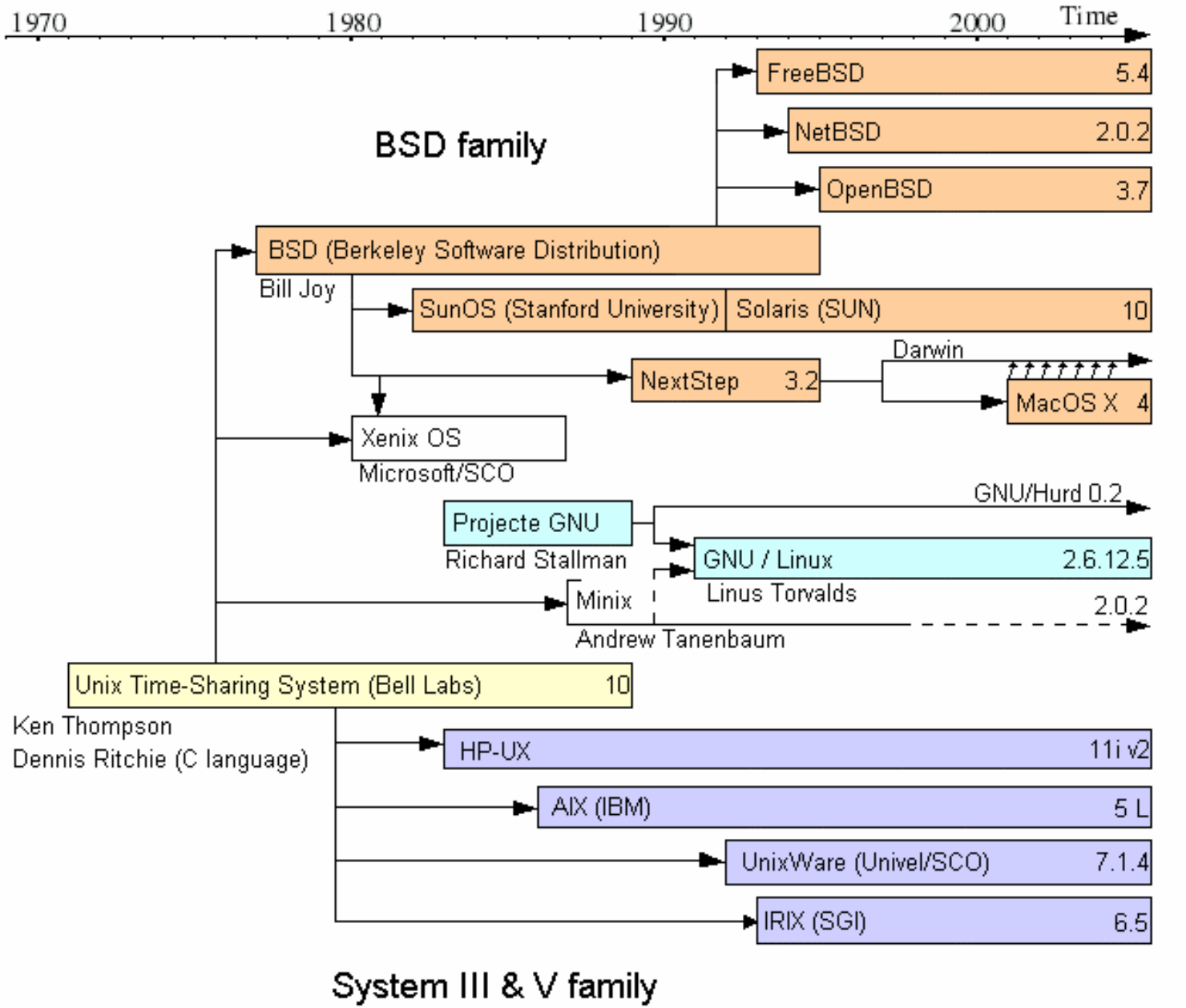
Unix kao način razmišljanja

Uvod u Unix

- UNIX - 1969 Thompson, Bell Laboratories, pisano u PDP-7 asm
- Thompson & Ritchie - prepisali u **C jezik**
- dizajn:
  - portabilno, višezadaćno, višekorisničko, vremensko dijeljenje resursa
- karakteristike:
  - obične tekstualne datoteke, CLI, hijerarhijski datotečni sustav, sve je datoteka (uređaji, IPC, itd.)

Uvod u Unix (2)

- SCO tužba:
  - 2000. Santa Cruz Operations prodaje UNIX poslove Caldera Systems, kasnije The SCO Group: besmislena SCO vs. Novell tužba
- **komercijalni** Unixoidi danas:
  - HP-UX, Solaris, Tru64, AIX, Mac OS X
- **slobodne** alternative:
  - FreeBSD, NetBSD, OpenBSD, DragonFly
  - Linux, OpenSolaris


1970
1980
1990
2000
Time
BSD family
FreeBSD 5.4
NetBSD 2.0.2
OpenBSD 3.7
BSD (Berkeley Software Distribution)
Bill Joy
SunOS (Stanford University)
Solaris (SUN) 10
Darwin
NextStep 3.2
MacOS X 4
Xenix OS
Microsoft/SCO
GNU/Hurd 0.2
Projecte GNU
Richard Stallman
GNU / Linux 2.6.12.5
Linus Torvalds
Minix
2.0.2
Andrew Tanenbaum
Unix Time-Sharing System (Bell Labs) 10
Ken Thompson
Dennis Ritchie (C language)
HP-UX 11i v2
AIX (IBM) 5 L
UnixWare (Univel/SCO) 7.1.4
IRIX (SGI) 6.5
System III & V family

Što je Unix

- 1994 Novell (dobio od AT&T) prenosi prava na UNIX tm i specifikaciju (kasnije Single UNIX Specification) na The Open Group (X/Open Company)
- također prodali i **kod** i **implementaciju** (UNIXWARE) SCO-u, koji je kasnije prenio tm UNIXWARE na The Open Group
- **definicija** se sastoji od nekoliko **standarda**: XPG4, POSIX i ISO C

Što je Unix (2)

- tijekom vremena su tm i specifikacija razdvojena, te omogućene višestruke implementacije
- Single UNIX **specifikacija** postala standard za UNIX sistemski API: UNIX 93, UNIX 95, UNIX 98 i UNIX 03
- no, postoje i Unixoidi!
- popularne konvencije: razlikujemo **UNIX** i **Unix**

Što je Unix (3)

- UNIX v7 - osnovna struktura - praktički standard:
  - **kernel**: conf (+boot), dev, sys, h (include)
  - **razvojno okruženje**: cc, as, ld, lib, include, ..., yacc, make, lex, ...
  - **naredbe**: sh, alati (sistemski: mkfs, fsck; administratorski: passwd, kill), za dokumente (troff, tbl, plot), komunikacija (mail, talk)
  - **dokumentacija**: man (manual stranice), doc (programski jezici)

Unix alternative

- 1983 RMS - GNU projekt:
  - **slobodan** OS nalik na Unix!
  - slobodan = svatko tko dobije kopiju bi trebao moći koristiti, proučavati, mijenjati i redistribuirati dotični
  - 1992: Linux jezgra pod GPL, GNU/Linux kao radeća implementacija jezgre - HURD nikad nije zaživio
  - GNU Compiler Collection, GNU C library, GNU core utilities
  - Red Hat, SuSE, Mandriva, Ubuntu, Debian, Gentoo, Knoppix, itd.

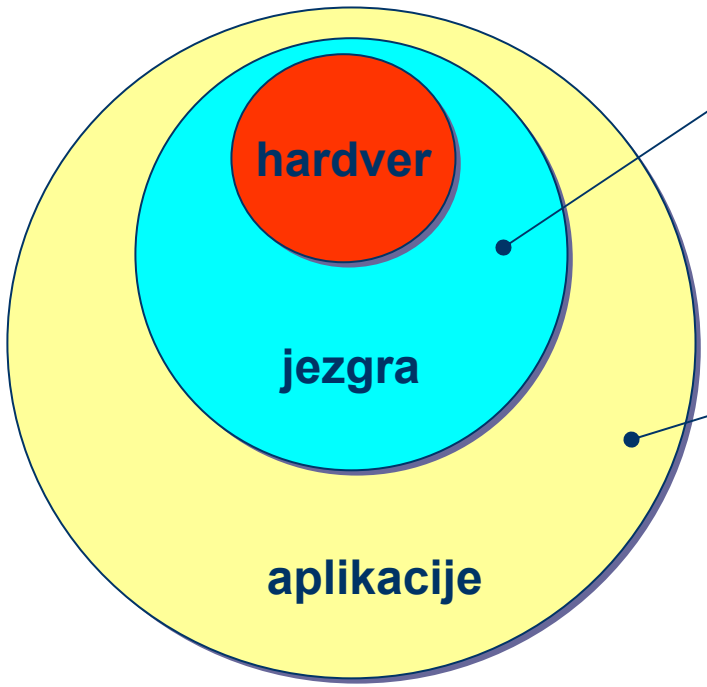
Unix alternative (2)

- 1994 - USL vs. BSDi:
  - tužba UNIX Systems Laboratories protiv University of California i Berkeley Software Design Inc.
  - Berkeley može distribuirati BSD Unix besplatno
  - OpenBSD, DragonFly BSD, FreeBSD itd.
- Apple - Mac OS X:
  - uzeo NEXTSTEP - pa preimenovano u Darwin
  - Mach mikrokernel, BSD + GNU alati/userland

Osnovne podjele

- operacijski sustav - sigurnija i kvalitetnija alternativa Windows poslužiteljima?
- osnovna podjela - dva dijela:
  - **kernel** (jezgra)
  - **userland** (programi)
    - shell (ljuska)
    - daemons (servisi)
    - X poslužitelj
    - programs/applications (korisnički programi)

Osnovne podjele (2)



jezgra sustava
(kernelland)

aplikacije - userland
međusobno također
strogo odvojene:
- ljuske
- razvojna okolina
- X11 i GUI
- servisi
- ostali korisnički programi
- igre
- itd.

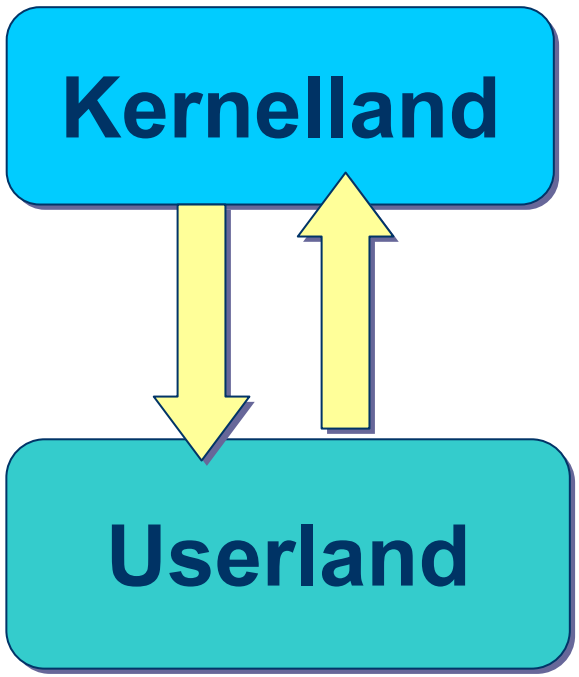
Kernel

- osnova operacijskog sustava - kontinuirano izvršavanje, uvijek - nije samostojeći!
- najniži dio softvera oko uređaja i hardvera
- programima omogućava pristup hardveru
- odlučuje kada, koliko i tko - multipleksiranje!
- pruža apstrakciju programima - skriva kompleksnost pristupa uređajima: HAL
- čisto i uniformno sučelje prema hardveru

Kategorije kernela

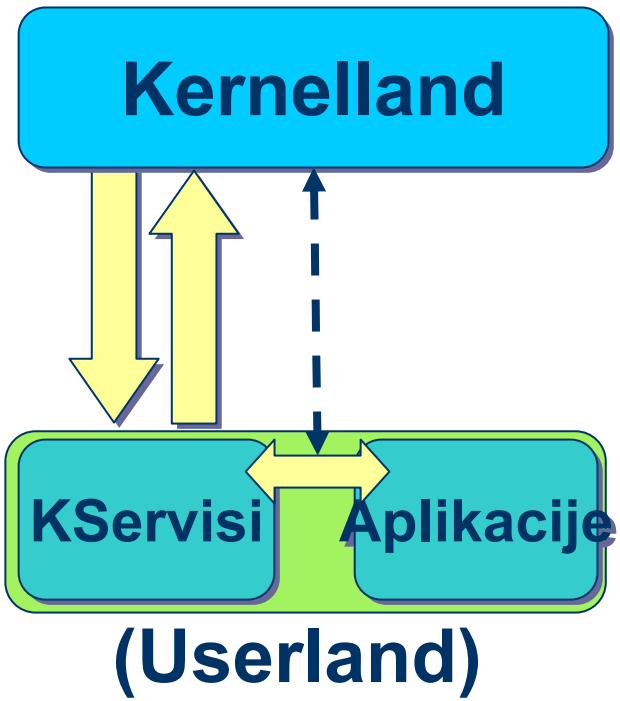
- **monolitni** - kompletna apstrakcija prema hw (standardni Linux, itd), zastarjelo!
- **mikrokernel** - minimalni funkcionalni set, pokretanje aplikacija (tzv. "serveri") za dodatne funkcionalnosti (L4/Fiasco, Mach, QNX)
- **hibridni** - dodatni kod na mikrokernel, brži rad (NT, BeOS, ReactOS, DragonFly)
- **eksokerneli** - biblioteke, nema apstrakcija

Monolitni kernel



- **naizgled** najjednostavniji
- vrlo **složene** apstrakcije
- **sistemski pozivi**: upravljanje procesima, konkurentnost, upravljanje memorijom
- niz **modula** - u **upravljačkom** načinu rada, obavljaju sistemske pozive, dijele istu memoriju, jaka integracija
- greška u **jednom** dijelu uzrokuje greške u **svim** dijelovima
- vrlo bliska **povezanost** koda na niskoj razini - dobra **efikasnost** i **brzina!**
- Linux, FreeBSD, Solaris, Windows NT, tradicionalni Unix (BSD obitelj)

Mikrokernel



- **minimalna**, jednostavna apstrakcija
- sistemski pozivi - **minimalni** OS **servisi**: upravljanje dretvama, adresnim prostorom i međuprocesna komunikacija
- ostale usluge kernela - **servisi** u **korisničkom** prostoru
- lako micanje nepotrebnih usluga
- **stabilnije** - gasi se samo jedan proces
- problem: nema garantiranog **stanja**!
- lošije **performanse** od monolitnih, mnogo **kopiranja** među aplikacijama/servisima (ctx switching)
- AIX, AmigaOS, Minix, QNX, Symbian, Mach: Hurd, Next/OpenStep, Mac OSX

Tru64 općenito

- nekad: Ultrix, OSF/1 AXP, Digital Unix
- **Alpha** osnovna hw platforma (planiran IA64)
- **verzije** danas:
  - 5.1B, 5.1A, 5.1, 4.0G, 5.0A, 5.0, 4.0F
- **značajke**:
  - 64-bit, za srednje i velike poslužitelje i mainframeove
  - SMP, clustering
  - max: 28GB memorije, 16TB FS & datoteka

Tru64 općenito (2)

- kernel:
  - 64-bit Mach 2.5 implementacija, mikrokernel
  - slično NeXTSTEP/Mac OS X
  - korijeni iz BSD 4.3/4.4, **System V**, itd.
- POSIX kompatibilno, RT podrška
- datotečni sustavi:
  - Berkeley UFS, AdVFS, NFS, ISO9660 (i RR ekstenzije), FAT, SMB, DFS

Tru64 budućnost

- ne baš bajna
- **2002** - HP kupio Compaq, najavio migraciju niza Tru64 mogućnosti u HP-UX
- krajem **2004** - odustalo se, otpuštaju se Tru64 developeri
- **2005** - HP planira prodavati proizvod kroz **2006** i sa završetkom godine prestati
- podrška planirana samo do **2011**, kao i AlphaServer hardver.

Tru64 linkovi

- Tru64 zajednica:
  - http://www.tru64.org/
- HP Tru64 program:
  - http://h30097.www3.hp.com/
- Zaštitite se:
  - http://sabernet.home.comcast.net/papers/Tru64.html
- Tru64 mailing liste:
  - http://www.ornl.gov/lists/mailing-lists/

Tru64 linkovi (2)

- Unix Rosetta Stone:
  - http://bhami.com/rosetta.html
- Usporedba OS-ova:
  - http://www.linuxrx.com/WS_Linux/OS_comparison.html
- stručni tehnički portali:
  - http://www.spyderbyte.com/
- Tru64 FAQ:
  - http://www.unixguide.net/compaq/faq/

Shell - ljuska

- dio operacijskog sustava, **program**
- **vanjski** dio sučelja između korisnika i operacijskog sustava
- **komandnolinijski** - CLI: sh, csh, tcsh, bash, zsh, ksh, cmd.exe, command.com
- **grafički** - GUI: Windows Explorer, Litestep, KDE, Gnome, CDE
- **sh** - Bourne shell, standard, /bin/sh
- shell programiranje, CLI naredbe..

Prvi pogled na Unix

- korisnički račun:
  - skup resursa potrebnih za rad
  - login, password, procesi, korisnički direktorij
  - UID = korisnička oznaka, GID = oznaka grupe
  - set osnovnih konfiguracijskih datoteka
  - izlazak sa sustava: naredbe logout, exit
  - pomoć na sustavu: naredba man (!!!)
  - promjena lozinke: naredba passwd

Man stranice

- sintaksa: man [section] title
- odjeljci:
  - 0 - ne koristi se
  - 1 - naredbe za sve korisnike
  - 1m - administratorske i ine naredbe
  - 2 - sistemski pozivi, sučelja jezgri i sl.
  - 3 - prog. sučelja u raznim bibliotekama
  - 4 - C zaglavlja, sistemske datoteke i sl.
  - 5 - razno (tekstualno procesiranje)
  - 6 - igre i sl
  - 7 - opisi uređaja, funkcionalnosti, mrežne podrške
  - 8 - naredbe za sistemsko održavanje

Dio II:
Upravljanje datotekama

Datotečni sustav - sve je datoteka

- apstrakcija implementacije za spremanje, organizaciju, manipulaciju, navigaciju, pristup i prihvat podataka
- spremišta mogu biti hard disk, CD, DVD, SAN, NAS, itd
- blokovi = sektori, 512bajtni
- jedna datoteka = 1 ili više blokova
- ime datoteke = simbolička važnost (ekstenzija, početak s točkom, itd)

Datotečni sustav (2)

- inode
  - struktura, osnovna informacija o datoteci/direktoriju/objektu
  - prema POSIX
    - **tip**: obična, direktorij, simbolički link, uređaj
    - oznaka **uređaja**
    - oznaka **korisnika** i **grupe**
    - **broj** koji determinira datoteku u dat. sustavu
    - **dozvole** - čitanje, pisanje, izvršavanje
    - **vremenska** oznaka - stvoren, promijenjen, pristupan
    - broj **referenci** - koliko čvrstih linkova pokazuje na inode

Datotečni sustav (3)

- tipovi
  - **diskovni** - fizički na uređaju, direktno ili indirektno:
    - FAT, NTFS, ext2, ext3, ISO9660, UFS, UDF, FFS, XFS
    - journaling - transakcije, sync, async, r/o, r/w, defragmentacija, volume manageri - LVM, versioning datotečni sustavi, RAID, itd.
  - **mrežni** - kroz mrežu, potencijalno dijeljeno:
    - SMB, CIFS, NFS, GFS, AFS
  - **baze** podataka - metapodaci, SQL:
    - Gnome storage, WinFS

Datotečni sustav (4)

- jedan osnovni direktorij
  - root direktorij, /
  - svaka datoteka na sustavu - unutar osnovnog
- nije nužno lokalan, može biti posve virtualan
- montiranje datotečnog sustava - sadržaj uređaja se postavlja "pod" neki direktorij
  - pri podizanju sustava - boot
  - ručno - mount, fstab
  - po potrebi - automount (mount i umount), supermount (nema sinkronizacije)

Datoteke

- sve je datoteka
- datoteka - obična, specijalna, direktorij
- ime - obično do 255 ASCII 7-bit znakova, bez specijalnih znakova
- "skrivena" datoteka - počinje sa .
- direktoriji - logička spremišta datoteka, odijeljena sa znakom /
- tip datoteke: naredba file

Hijerarhija

- osnovni direktorij = /
- poddirektoriji odvojeni sa /
- standardne konvencije:
  - najčešći prefiksi:
    - /, /usr, /usr/local, /opt, /usr/ucb, /usr/ccs
  - najčešći direktoriji - standardno se zna namjena pojedinih:
    - bin, sbin, include, sys
    - home, var, tmp, spool
- staze: apsolutne i relativne

Hijerarhija (2)

- root fs = /
  - /usr
    - /usr/sbin - administr. izvršne datoteke, dinamički linkane
    - /usr/bin - standardni alati za korisnike
    - /usr/ccs - C compiler i popratni alati u bin direktoriju, C biblioteke i sl. u lib direktoriju, C zaglavlja u include, ...
    - /usr/users - korisnički direktoriji!
    - /usr/lbin - backend izvršne datoteke
    - /usr/opt - opcionalni dodatni alati
    - /usr/share - platformski nezavisne različite ASCII datoteke: man stranice, makroi, rječničke liste, terminfo

Hijerarhija (3)

  - /usr/shlib - dijeljene biblioteke za dinamički linkane programe
  - /usr/sys - konfiguracijske datoteke od kernela
- /sbin - naredbe i programi nužni za podizanje sustava
  - /sbin/init.d - sistemske rc skripte za podizanje/spuštanje servisa
  - /sbin/rc0.d, /sbin/rc1.d, /sbin/rc2.d
- /opt - dodatne instalirane aplikacije i uređaji, obično komercijalni

Hijerarhija (4)

- /dev - uređaji (znakovni i blokovski) u vidu specijalnih datoteka
- /devices - isto (noviji Tru64)
- /etc - konfiguracijske datoteke i različite baze za sustav
  - /etc/nls - lokalizacijske datoteke
- /lost+found - datoteke koje je fsck oporavio
- /vmunix - jezgra sustava
- /cluster - direktorij za cluster u kojem se nalazi specifični čvor (member0)

Hijerarhija (5)

- /tmp - privrem. datoteke, nestaju nakon reboota
- /var - logovi, privremene datoteke, datoteke za zaključavanje i sl.
  - /var/opt - opcionalno...
  - /var/tmp - privremene datoteke
  - /var/spool - spooling (mail, printer datoteke)
  - /var/adm - administrativne datoteke i baze: npr. crash za crashdumpove, cron za crontab datoteke, sendmail za sendmail konfiguraciju, syslog za logove
- /subsys - dinamički kernel moduli
- /sys - symlinkovi na source u /usr/sys

Dozvole

- restrikcije pristupa (čitanja, pisanja i izvršavanja) nad datotekama i direktorijima
- tri **grupe** dozvola nad svakom datotekom:
  - u = korisnik, g = grupa, o = ostali
- moderniji Unixi - extended ACL
- tri **tipa** dozvola:
  - r = čitanje - čitanje datoteke, listing direktorija
  - w = pisanje
  - x = izvršavanje - izvršavanje datoteke, ulazak u direktorij

Dozvole (2)

- prikaz dozvole - naredba ls:
  - `ls -al datoteka (ili direktorij)`
  - `-rwxr-xr-x 1 erpl08 staff 3649 Feb 22 15:51 my.html`
  - redom:
    - korisničke dozvole, grupne dozvole, other dozvole
    - broj linkova
    - ime vlasnika
    - ime vlasničke grupe
    - vrijeme i datum zadnje promjene
    - ime datoteke

Dozvole (3)

- standardne Unix dozvole:
  - rudimentarno grupiranje korisnika (korisnik/grupa/ostali)
  - rudimentarno upravljanje: nema nasljeđivanja, nema iznimaka, nema listi dozvola po različiti korisnicima "per file"...
  - execute ima višestruka značenja - direktorij, datoteka
  - dodatni bitovi: sticky, setuid, setgid...
  - oprez! lako se griješi...

Kretanje po direktorijima

- hijerarhijska struktura:
  - kretanje do traženog direktorija (apsolutno ili relativno)
- koncepti:
  - **korisnički direktorij** (home directory): pravo pisanja, vlastiti direktorij, $HOME, razne dot-rc (konfiguracijske) datoteke, privremene datoteke
  - **tekući** direktorij (working directory): radni/trenutni direktorij u hijerarhiji, $PWD, naredba pwd za ispis

Kretanje po direktorijima (2)

- staza (pathname):
  - direktoriji i poddirektoriji do željenog mjesta (datoteke ili direktorija)
  - nužno je objasniti naredbama i programima lokaciju objekta
  - **apsolutne** staze - puna staza unutar cijele hijerarhije, uvijek jedinstveno opisuje odredišni objekt: npr. `/usr/local/bin/ls`
  - **relativne** staze - pomak od trenutnog direktorija, npr. `ls .././pero.txt`

Kretanje po direktorijima (3)

- naredba cd:
  - . je tekući direktorij
  - .. je prethodni direktorij
  - / je osnovni direktorij
- primjeri:
  - `cd; cd book; cd /usr/users; cd ..`
  - `pwd`
- za ulazak u direktorij nužna x (izvrši) dozvola

Radni zadatak

- prikažite sadržaj datoteke /etc/passwd na ekranu koristeći naredbu cat
- naredba cat prima kao argument stazu do datoteke, bilo relativnu bilo apsolutnu
- napravite gornji zadatak koristeći:
  - apsolutno referenciranje datoteke
  - relativno referenciranje datoteke
  - šetnjom kroz direktorije
- na kraju zadatka vratite se u korisnički direktorij (home)

Ispisivanje liste datoteka

- naredba ls prikazuje informacije o datotekama i direktorijima, ako je to moguće (ovisi o r i x dozvolama)
- osnovni parametri:
  - -a - prikazuje sve datoteke, čak i one sa .
  - -F - klasificira datoteke (specijalni simboli za specijalne datoteke)
  - -l - dugi, temeljitiji ispis
  - -R - rekurzivni ispis

Ispisivanje liste datoteka (2)

- primjeri:
  - `ls; ls /usr/users; ls .; ls ..; ls pero zdero`
  - `ls -al`
  - `ls -alR`
  - `ls -aF`
- za rekurzivni ispis:
  - nužne x (**izvrši**) dozvole za ulazak u direktorij
  - standardno nužne r (**čitaj**) dozvole za obični ispis datoteka

Radni zadatak

- pozicionirati se u vlastiti korisnički direktorij
- ispisati sve datoteke u tekućem direktoriju (.) i direktoriju iznad njega (..)
- promijenite direktorij u osnovni (root odnosno /) direktorij i ispišite sadržaj /bin i /etc direktorija
- promijenite direktorij u /usr/bin, te sa tog mjesta ispišite sadržaj /usr direktorija i vlastitog korisničkog direktorija
- vratite se u vlastiti korisnički direktorij

Pronalaženje datoteka

- naredba whereis:
  - -b - traži izvršne datoteke (programi)
  - -m - traži manual stranice (priručnici)
  - -s - traži izvorne datoteke

- primjeri:
  - `whereis ls; whereis -b ls`
  - `whereis -m ls; whereis -s ls`

Pronalaženje datoteka (2)

- naredba find:
  - upotreba: find staza izraz
  - izraz:
    - -ls - ispiši u dugoj formi (puno informacija)
    - -print - ispiši standardno
    - -exec program "{}" ";" - izvrši program nad svakom datotekom
    - -size veličina - traži veličinu
    - -name izraz2 - traži datoteku koja odgovara izrazu
    - -user korisnik - ... čiji je vlasnik rečeni korisnik
    - -group grupa - ... čiji je vlasnik rečena grupa
    - -atime n - ... pristupano prije n dana

Pronalaženje datoteka (3)

- primjeri:
  - `find /usr/users -name .profile -print`
  - `find / -user root -ls`
  - `find / -atime 7 -ls`
  - `find . -type f -exec grep -i pero "{}" ";" -print`
  - `find / -size 0 -print`
  - `find / -atime 3 -ls`
  - `find . -type f -size 0 -exec rm "{}" ";"`

Pronalaženje datoteka (4)

- -name izraz2 (pattern matching):
  - izraz2 može sadržavati na bilo kojem mjestu:
    - * - mijenja niz znakova: "*nanas" ili *nanas; "auto*" ili auto*; itd.
    - ? - mijenja jedan znak: "?abuka" ili \?abuka; itd.

- primjeri:
  - `find . -name "*pero*" -print`
  - `find . -name "*.tmp" -exec rm "{}" ";" -print`

Pronalaženje datoteka (5)

- tipična upotreba:
  - pronalaženje setuid/setgid datoteka na sustavu
  - brisanje mnogo sličnih datoteka (zajednički podniz ili neki drugi zajednički kriterij) unutar nekog direktorija
  - ispis datoteka koje zadovoljavaju nekakav zajednički kriterij
  - backup ili neka proizvoljna akcija nad sličnim datotekama...

Radni zadatak

- iz vlastitog korisničkog direktorija...
- pretražite /etc direktorij u potrazi za datotekom motd i ispišite njene dozvole
- potražite /usr/users direktorij za svim datotekama koje posjedujete i ispišite ih (koji ste korisnik možete utvrditi naredbom id)
- pretražite /dev direktorij za datotekama koje sadrže niz "tt" i ispišite ih u dugoj formi

Čitanje datoteka

- naredbe:
  - cat - **ispisuje** tekst na konzolu bez **prekida**
  - more - **ispisuje** po jednu **stranicu** ili liniju i omogućava osnovno kretanje
  - head - **ispisuje** 10 ili proizvoljni broj **linija** sa početka
  - tail - **ispisuje** 10 ili proizvoljni broj **linija** sa kraja
  - sve primaju jednu ili više datoteka u formi relativne ili apsolutne staze kao argument

Čitanje datoteka (2)

- naredba head:
  - argumenti: -n brojlinija ili (zastarjelo) -brojlinija
  - argumenti: -c brojznakova
  - primjer: `head -n 6 /etc/passwd`
- naredba tail:
  - argumenti: -n +-brojlinija ili (zastarjelo) +-brojlinija
  - argumenti: -c +-brojznakova
  - primjer: `tail -4 /etc/passwd`
  - primjer: `tail +1 /etc/passwd`

Čitanje datoteka (3)

- naredba more:
  - tipka enter - iduća linija
  - broj pa enter - prikazuje se odgovarajuće mnogo idućih linija
  - tipka space - prikazuje se slijedeći ekran linija
  - tipka b - prikazuje se prethodni ekran linija
  - tipka q - izlazak u ljusku
  - tipka h - za pomoć
  - tipka / - za pretraživanje teksta
  - tipka n - za ponovno pretraživanje upisanog niza

Čitanje datoteka (4)

- tipična upotreba:
  - less/more za pregled logova i datoteka interaktivno, ali bez editiranja
  - tail za pregled najsvježijih događaja u sistemskim zapisnicima
  - head za pregled najstarijih događaja...
  - tail za ispis događaja "uživo"!

Radni zadatak

- iz vlastitog korisničkog direktorija ispišite datoteke u /etc direktoriju, potražite datoteku imena passwd
- ispišite sadržaj datoteke passwd na ekranu
- ispišite je ekran po ekran
- ispišite prvih 6 linija
- ispišite zadnjih 9 linija

Stvaranje datoteka

- naredba touch:
  - stvara **praznu** datoteku
  - **mijenja vrijeme** pristupa i promjene datoteke
  - parametri:
    - ništa - mijenja vremena na datoteci na trenutno ili je stvara
    - -a - samo pristupno vrijeme
    - -m - samo vrijeme promjene
    - -r - koristi neku drugu datoteku kao referencu
    - -t mijenja vrijeme modifikacije koristeći ručno zadano vrijeme: [[CC]YY]MMDDhhmm[.SS]

Stvaranje datoteka (2)

- primjeri:
  - `touch pero.c`
  - `touch -r /etc/passwd pero.cpp`

- naredba cat:
  - služi za **ispis** datoteka
  - stvaranje datoteka i dodavanje sadržaja
  - spajanje više datoteka u jednu
  - kombinacija tipki **ctrl d** - označava EOF

Stvaranje datoteka (3)

- osnovno **preusmjeravanje**:
  - > - ljuska **briše** stari sadržaj i upisuje **novi**
  - >> - ljuska **dodaje** novi sadržaj na stari

- primjeri:
  - `cat > pero`
  - `cat >> pero`
  - `cat pero`
  - `cat pero /etc/passwd > nova`
  - `cat nova >> pero`

Radni zadatak

- u vlastitom korisničkom direktoriju (provjerite!) stvorite datoteku voce sa 10ak vrsta voca, te u istom direktoriju stvorite datoteku povrce sa par vrsta povrca
- spojite te dvije datoteke (sadržaj datoteke povrce dodajte na sadržaj datoteke voce) i ispišite na ekran
- pronađite datoteku fstab u /etc direktoriju i koristeći nju promijenite vremena dotične datoteke i provjerite točnost (ispišite obje!)

Stvaranje direktorija

- logični korak
  - organizacija datoteka po direktorijima
  - smanjenje opterećenja! - dirent() poziv "skup" u prevelikim direktorijima (mnogo datoteka)
- mkdir naredba:
  - stvara direktorij, ako nema datoteke sa istim imenom
  - prima apsolutnu ili relativnu stazu
  - najčešće u kombinaciji sa naredbom cd
  - primjeri: `mkdir pero zdero`

Radni zadatak

- stvorite vjezba direktorij u vašem korisničkom direktoriju
- ispišite sve datoteke u dotičnom, bez mijenjanja tekućeg direktorija
- također bez promjene tekućeg direktorija napravite vjezba1 i vjezba2 direktorije u vjezba direktoriju
- ispišite rekurzivno sve datoteke unutar vjezba direktorija

Mijenjanje dozvola

- kratki repetitorij...
- dozvole - restrikcija pristupa datotekama i direktorijima
- tri osnovna **tipa** dozvola - **r**, **x** i **w**
- tri **grupe** dozvola - **u**, **g**, **o**: rwx rwx rwx
- primjer:
  - `bash-2.05$ ls -al voce`
  - `-rw-r--r--  1 dkorunic nis 21 Jan 26 13:40 voce`

Mijenjanje dozvola (2)

- prvo polje - **tip datoteke**:
  - - - obična datoteka
  - d - direktorij
  - c - specijalni znakovni uređaj
  - P - imenovani cjevovod
  - s - socket
  - l - simbolički link
- izvršna datoteka - N.B. direktorij!:
  - program ili skripta
  - mora imati x postavljen

Mijenjanje dozvola (3)

- naredba chmod:
  - prima oktalni broj: r = 4, w = 2, x = 1
  - zbrajaju se željeni brojevi za dozvolu
  - primjer: `chmod 660 pero.txt`
  - prima i slovni izraz:
    - + - dodaj dozvolu,
    - - - oduzmi dozvolu,
    - = - promijeni na novo
    - ...

Mijenjanje dozvola (4)

- u = user, korisničke dozvole
- g = group, grupne dozvole
- o = other, dozvole svima ostalima
- a = dozvole svima
- standardne r/w/x oznake,
- znak , za separator u izrazu

– primjer: `chmod ugo=x pero.txt`
– primjer: `chmod u+rwx pero.txt`
– primjer: `chmod g-r,o+r pero.txt`

Radni zadatak

- u vlastitom korisničkom direktoriju napravite datoteku grupa sa sadržajem datoteke /etc/group
- prikažite dozvole na jednoj i drugoj datoteci
- koristeći naredbu chmod (kako oktalne tako i slovne oznake) promijenite dozvole na grupa datoteci da budu: `rwx rw- --x`
- promijenite dozvole (također i oktalno i simbolički) tako da korisnik ima rw, te da grupa i ostali nemaju pristupa

Mijenjanje dozvola (5)

- (uglavnom) potrebne root dozvole:
- naredba chown:
  - mijenja vlasnika datoteke ili direktorija
  - opcionalno mijenja i grupu zajedno sa vlasnikom
  - primjer: `chown dkorunic pero.txt`
  - primjer: `chown dkorunic:nis pero.txt`
- naredba chgrp:
  - mijenja grupu datoteke ili direktorija
  - primjer: `chgrp nis pero.txt`

Micanje i preimenovanje datoteka

- naredba mv:
  - mv dat1 dat2
  - mv dat ... direktorij
  - fizički se vrši kopiranje, već se samo **mijenja pokazivač** na datoteku - operacija je vrlo efikasna
  - primjer: `mv pero zdero`
  - primjer: `mv a b direktorij`
  - primjer: `mv a direktorij/b`
  - primjer: `mv direktorij/* .`
  - mv **briše** stari sadržaj datoteke **bez pitanja**!

Radni zadatak

- u vašem direktoriju napravite datoteku lista sa ispisom svih datoteka u direktoriju (koristeći > kao redirekciju)
- promijenite datoteci lista ime u listurina
- napravite direktorij temp
- pomaknite datoteku listurina u direktorij temp i istovremeno joj promijenite ime nazad u lista, koristeći samo jednu naredbu

Kopiranje datoteka

- naredba cp:
  - stvara **novu datoteku** sa sadržajem originalne
  - originalna je **netaknuta**
  - standardno neće očuvati vrijeme pristupa i promjene
  - moguće kopirati cijele direktorije
  - parametri:
    - -p - očuvati će sve što može (promjenu, pristup, dozvole, korisnika, grupu, itd.)

Kopiranje datoteka (2)

- parametri:
  - -R - rekurzivno kopiranje direktorija, pazeći da očuva istovjetnost (simbolički link ostaje takav, FIFO ostaje takav, itd)
  - -r - rekurzivno kopiranje direktorija, kopira sadržaj, neće očuvati specijalne datoteke
- primjeri:
  - `cp pero /dev/null`
  - `cp -R radni drugi_direktorij`
  - `cp * ../c`
  - `cp /etc/group .`

Radni zadatak

- tekući direktorij je korisnički direktorij
- skopirajte /etc/passwd u vlastiti direktorij, pod imenom lozinka1
- napravite direktorij vjezba
- skopirajte sadržaj lozinka1 u vjezba direktorij, ne mijenjajući tekući direktorij
- premjestite sav (dakle koristeći *) sadržaj vjezba direktorija u korisnički direktorij

Brisanje datoteka i direktorija

- naredba rmdir:
  - briše direktorij samo ako je **prazan**
  - rjeđe se koristi
  - primjer: `rmdir prazni1 prazni2`
- naredba rm:
  - briše datoteke i direktorije
  - provjerite! pažljivo!
  - parametri:
    - -f - forsira brisanje, neovisno o dozvolama i sadržaju
    - -r - briše rekurzivno zadane datoteke i/ili direktorije

Brisanje datoteka i direktorija (2)

- primjer: `rm prva druga treca`
- primjer: `rm -r direktorij`
- primjer: `rm -rf dat1 dir1 dat2 dir2 dir3`
- primjer: `rm -rf *`
- primjer (ne moraju postojati): `rm -f pero* a*`

- sadržaj nepovratno izgubljen, osim ako ne postoji **hardlink**
- ako se obriše datoteka na koju pokazuje **softlink**, softlink više ne vrijedi (pokazuje krivo!) - više o tome kasnije!

Radni zadatak

- u korisničkom direktoriju stvorite direktorije vjezba1 i vjezba2
- promijenite tekući direktorij u vjezba1
- napravite datoteku prijatelji sa imenima vašim prijatelja ili poznanika
- napravite 3 kopije iste datoteke: prijatelji1, prijatelji2 i prijatelji3
- obrišite datoteku prijatelji1
- vratite se u direktorij iznad i obrišite cijeli direktorij vjezba1, a zatim i vjezba2

Povezivanje datoteka

- više datoteka - samo jedan fizički sadržaj
- dva osnovna tipa linka:
  - **simbolički** - symlink
    - samo kao pokazivač, očigledan u ispisu direktorija
  - **čvrsti pokazivač** - hardlink
    - izgleda kao standardna datoteka
    - pokazuje na inode
    - brisanjem originalne datoteke sadržaj ne nestaje, već je dobavljiv kroz hardlink
    - ne može pokazivati na direktorij
    - mora biti na istom datotečnom sustavu!

Povezivanje datoteka (2)

- riješeno vrlo jednostavno - inodeovi su bitni
- naredba ln:
  - važno: ln originalna_dat [odredišni_link]
  - bez drugog argumenta podrazumijeva isto ime
  - parametri:
    - -f - forsira brisanje originalnog sadržaja
    - -s - stvara simbolički link
    - inače stvara hardlink
  - primjer: `ln -s /etc/passwd datoteka`
  - primjer: `ln /etc/group /etc/group-back`

Povezivanje datoteka (3)

- tipične primjene:
  - softlink:
    - jedan originalni direktorij, više symlinkova na njega...
    - biblioteke - različite instalirane verzije, jedna aktualna (posljednja verzija!)
    - jednostavno preusmjeravanje, npr. u /dev/null za neku log datoteku
  - hardlink
    - instalacija više datoteka ali jednog sadržaja (hmm)
    - garantirano postojanje sadržaja na tom fs-u, očuvanje sadržaja i u slučaju brisanja jednog od pokazivača: npr. za backup

Radni zadatak

- budite u korisničkom direktoriju
- napravite simbolički link na /etc/passwd
- napravite datoteku pero sa proizvoljnim sadržajem
- napravite hardlink pero2 na pero
- pregledajte ispis i sadržaje stvorenih linkova
- napravite simbolički link na /etc direktorij
- uđite u simbolički link koji pokazuje na /etc

Dio III:
UNIX ljuske

Ljuska

- sučelje prema jezgri sustava, shell
- samostojeći interpreter
  - prevodi i izvršava pojedinačne naredbe, interne i eksterne
  - izvršava programe i brine se o statusu te omogućava upravljanje "poslovima"
  - omogućava prikaz na konzoli
  - omogućava izvršavanje shell skripti
- vrste: C shell (csh), Bourne shell (sh), Korn shell (ksh)

Ljuska (2)

- kako pronaći koja je tekuća ljuska:
  - naredba ps
  - pregledom /etc/passwd (grep...)
- promjena ljuske:
  - naredba chsh
  - samo ako je ljuska navedena u /etc/shells
  - vrijedi nakon ponovnog spajanja
- privremena promjena:
  - pozivanjem odgovarajuće ljuske
  - primjer: `/bin/csh; /bin/bash`

Zamjenski znakovi

- znakovi sa specijalnim značenjem za ljusku!
- zamjena nula, jednog ili više znakova
- grupne operacije (cp, mv...)

- operator .:
  - mijenja isključivo 1 znak
  - ako ime datoteke počinje sa točkom, koristi se
  - taj znak smije biti i prijelaz u idući red "\n"
  - primjer: `.*file`

Zamjenski znakovi (2)

- operator *:
  - mijenja 0 ili više znakova
  - neće uspjeti zamijeniti datoteke koje počinju sa . - skrivene datoteke!
  - primjer: `file*` - počinje sa nizom file
  - primjer: `*.c` - završava sa nizom .c
  - primjer: `*f*` - sadržava znak f
  - primjer: `fi*.t` - počinje sa fi i završava sa .t

Zamjenski znakovi (3)

- operator ?:
  - mijenja isključivo 1 znak
  - primjer: `file?` - file i samo 1 proizvoljan znak
  - primjer: `???` - isključivo 3 znaka
  - primjer: `file*.?` - sufiks je proizvoljan niz znakova, točka i još jedan znak
- operator [...]:
  - mijenja bilo koji znak koji je naveden u grupi
  - primjer: `[a-d]`
  - primjer: `[A-Z]`

Zamjenski znakovi (4)

- operator [!...]:
  - mijenja bilo koji znak koji nije naveden u grupi
  - niz počinje uvijek sa !
- onemogućiti specijalno značenje:
  - koristeći jednostruke navodnike, npr. '*niz*', '?a'
  - koristeći escaping, npr. *niz*, \?a
- direktoriji i datoteke ne bi smjeli sadržavati zamjenske znakove!

Radni zadatak

- koristeći zamjenske znakove, ispišite sve datoteke iz /etc direktorija koje završavaju sa .conf, a zatim one koje počinju sa .new
- stvorite datoteku naziva ? u korisničkom direktoriju i obrišite ju
- koristeći zamjenske znakove, ispišite sve datoteke iz /etc direktorija koje sadržavaju .rc. niz znakova i barem po 1 neprazni znak prije i poslije .rc. niza

Sh

- paraleliziranje, ulančavanje - pipeline
- odvojene sa | ili ^ (zastarjelo, nestandardno)
- mogući razdjelnici:
  - ; - sekvencijalno izvršavanje, nema stvarnog povezivanja
  - & - asinkroni rad
  - && - logički and
  - || - logički or
- detaljnije u nastavku...

Sh - kontrola toka

- izvršavanje liste naredbi za svaku riječ:
  - `for identifier [in word ...] do list done`
- ako riječ odgovara uzorku, izvršava listu:
  - `case word in [pattern [ | pattern ] ...) list ;;] ... esac`
- ako se uspješno izvrše naredbe iz liste, onda izvršava prvu listu, u suprotnom iza elif odnosno else...:
  - `if list then list [elif list then list] ... [else list] fi`

Sh - kontrola toka (2)

- izvršava list2 dokle god list1 vraća 0:
  - `while list1 do list2 done`
- izvršava list2 dok list1 ne vrati 0:
  - `until list1 do list2 done`
- izvršava list u podljusci:
  - `(list)`
- izvršava u trenutnoj ljusci:
  - `{ list; }`
- definira funkciju:
  - `name () { list; }`

Sh - pozicionalni parametri

- shell skripte, funkcije
- riječ na 0 poziciji (ime funkcije, skripte, procedure) je $0
- riječ na 1 (prvi parametar) je $1
- itd.
- nužna naredba shift za dobivanje argumenata većih od $9
- svi argumenti: $@
- primjer: `echo $0`

Sh - varijable

- pridjeljivanje: ime=vrijednost
- dohvat sadržaja - prefiksiranje sa $
- niz obuhvaćen sa "" (dvostruki) ili '' (jednostruki navodnici) - razmaci, tabovi, točkazarez i prijelaz u novi red netaknuti
- "" - ima supstitucije za varijable, za '' nema!
- primjer:
  - `pero=bla; echo $pero`
  - `pero="$pero$pero"; pero='$pero$pero'`

Sh - varijable (2)

- međutim, nekad je nužno koristiti ${ime}
  - `tezina=10`
  - `echo "$tezina kilograma"`
  - `echo "${tezina}kg"`
- moguće definirati samo za jednu naredbu:
  - `a=3 b=4 echo $a $b`
  - `echo $a $b`
- naredba env:
  - ispisuje sve varijable okoline

Sh - specijalne varijable

- $# - broj pozicionalnih parametara
- $? - izlazna vrijednost zadnjeg programa
- $$ - PID trenutnog procesa ljuske
- $! - PID zadnjeg procesa u pozadini (operator &)
- $@ i $* - svi pozicionalni parametri kao jedan niz znakova
- primjer:
  - `echo $?; touch pero.$$`

Sh - hvatanje izlaza naredbe

- koriste se obrnuti navodnici ``
- primjer:
  - `pero=`ls``
  - `datoteke=`ls | wc -l``
- više u poglavlju o ulančavanju i preusmjeravanju ulaza i izlaza!

Sh - ugrađene naredbe (neke...)

- exit - izlaz iz trenutne ljuske
- export - označi varijablu za prijenos okoline
- eval - svoje argumente izvršava kao naredbu
- echo - ispis teksta
- exec - izvrši umjesto trenutne ljuske
- read - čita red po red i pridružuje varijablama
- test - provjerava ispravnost izraza
- shift, pwd, cd... i mnoge druge...

Ksh i okolina

- ksh - Korn shell
- vrijedi za većinu modernih ljuski
- roditelj-ljuska predaje neke osobine djetetu (naslijeđena okolina):
  - prava izvršavanja i pokretanja
  - datoteke koje je otvorio roditelj
  - dozvoljene limite u sustavu
  - postavke kontrole signala
  - vrijednosti IFS varijable

Ksh i okolina (2)

- roditelj u svojoj okolini ima još neke osobine, koje se ne predaju djetetu:
  - aliasovi - zamjenski nazivi
  - funkcije - dijelovi koda
  - varijable
  - vrijednosti rezerviranih varijabli
  - različite postavke i opcije ljuske

Ksh - varijable

- automatske varijable - postavlja ih interno ljuska, ovisno o programima i radu:
  - ERRNO - broj zadnje greške (neuspjelog poziva)
  - LINENO - broj linije koja se izvršava u skripti/funkciji
  - OLDPWD - prethodni radni direktorij
  - PPID - PID roditelja
  - RANDOM - pseudoslučajni broj između 0 i 32767
  - SECONDS - broj sekundi od pokretanja ljuske
  - $ - PID trenutnog procesa

Ksh - varijable (2)

  - ! - PID zadnjeg procesa u pozadinskom izvršavanju
  - ? - status izvršavanja zadnje naredbe, programa: 0 znači uspješno izvršavanje, sve što nije 0 je oznaka greške koju je program/naredba vratila
- neautomatske - postavljaju se ručno, daju dodatne funkcionalnosti, koriste ih neki programi ili sama ljuska
  - CDPATH - lista odvojena sa :, koristi se za cd naredbu i pretraživanje željenog direktorija

Ksh - varijable (3)

- EDITOR - staza i naredba do željenog CLI editora
- ENV - ime skripte koja se izvršava svaki put kad se poziva ksh
- FCEDIT - editor za fc naredbu (editiranje povijesti ljuske)
- HISTFILE - datoteka za spremanje povijesti ljuske
- HISTSIZE - broj mogućih linija u HISTFILE
- HOME - vlastiti korisnički direktorij
- MAIL - staza do e-mail sandučića

Ksh - varijable (4)

- MAILCHECK - koliko će često u sekundama ljuska provjeravati ima li nove pošte
- PATH - lista odvojena :, predstavlja set staza koje će se pretraživati u potrazi za zadanom naredbom
- PS1 - izgled prompta ljuske
- PS2 - izgled prompta ljuske kad je duža od 1 reda
- TERM - tip terminala koji se trenutno koristi
- VISUAL - identično EDITOR
- DISPLAY - označava put do X terminala (GUI)

Ksh - upravljanje varijablama

- pristup varijabli
  - kroz operator $ za saznavanje sadržaja i odgovarajuću naredbu
  - naredba print (zastarjelo!)
  - naredba echo
  - naredba set - ispisuje sve varijable
  - primjer: `print $EDITOR`
  - primjer: `echo $EDITOR $MAILCHECK`
  - primjer: `set`

Ksh - upravljanje varijablama (2)

- postavljanje:
  - varijabla=vrijednost (bez razmaka!)
  - primjer: FCEDIT=vi
  - primjer: EDITOR=vi
- zaboravljanje varijable:
  - naredba unset
  - primjer: unset EDITOR FCEDIT

Ksh - upravljanje varijablama (3)

- postavljanje tipova varijabli i/ili vrijednosti:
  - naredba typeset
  - parametri:
    - -u - velika slova
    - -l - mala slova
    - -in - cjelobrojno u bazi n
    - -r - samo se može čitati
    - -x - varijabla se prenosi
  - rjeđe se koristi

Ksh - konfiguriranje

- postoje razne interne varijable
- mijenjanju način rada, dodaju ili oduzimaju mogućnosti
- naredba set:
  - -o - prikazuje sve mogućnosti
  - +o opcija - postavlja dotičnu mogućnost
  - -o opcija - gasi dotičnu mogućnost
- obično se samo jednom u životnom ciklusu konfigurira ljuska, nakon toga se troši postojeća konfiguracija

Ksh - konfiguracijske datoteke

- standardno:
  - /etc/profile - prva se izvršava
  - $HOME/.profile - zatim ova, ako postoji
  - $ENV - npr. $HOME/.kshrc
- služi za:
  - postavljanje vlastitih varijabli i parametara
  - definiranje skripti, aliasova i funkcija
  - primjer: `alias lal='ls -al'`
  - primjer: `HISTSIZE=64`

Radni zadatak

- provjerite koju ljusku imate
- pregledajte koje varijable ima postavljena vaša ljuska
- ispišite vrijednost $EDITOR, $HOME i $PATH varijabli
- pokušajte izbrisati $PATH varijablu - što će se dešava?

Dio IV:
Vi editor

Unix editori

- ed - linijski editor, radi u single user načinu rada, jedva upotrebljiv
- ex - ed-oliki editor, više mogućnosti
- sed - linijski editor toka, isključivo za prompt, i shell skripte
- vi - pravi editor, vrlo mnogo mogućnosti (danas postoji odlični vim)
- emacs - vrlo moćni editor današnjice, ali vrlo težak za početnike (danas xemacs)

Uvod

- bitno da je $TERM ispravna: vt100, ansi, itd.
- pozivanje: vi [datoteka]
- osnovna tri načina rada:
  - komandni
    - standardni koji je aktivan po ulaženju u editor
    - omogućava pomicanje po tekstu, brisanje teksta i undo naredbe
    - iz tekst moda se u njega dolazi sa esc tipkom
    - esc prekida dio naredbe
    - npr. ZZ (izlazak iz editora)

Uvod (2)

- tekstualni
  - unos teksta
  - aktivira se nakon a, A, c, C, i, I, o, O, R, s, S naredbi
- zadnja linija - prošireni komandni:
  - aktivira se iz komandnog na unošenje :, /, ? ili ! znakova (započinju specijalne naredbe)
  - editiranje druge datoteke, izvršavanje ex naredbe, izvršavanje naredbe ljuske, izlazak, čitanje nove datoteke, pisanje sadržaja u datoteku, itd
  - završava sa enter tipkom
- konvencija: prazne linije se prikazuju sa znakom ~

Izvršavanje i izlaženje

- :wq - izlaženje uz snimanje promjena
- ZZ - isto kao gornji primjer
- :q - izlaženje ako nije bilo promjena
- :q! - izlaženje bez obzira na promjene, sa zaboravljanjem promjena
- :w !naredba - izvršava datoteku kroz naredbu bez promjene po sadržaj (oprez zbog :w!)
- :!naredba - izvršava naredbu
- :sh - izvršava ljusku

Pomicanje kursora

- kursor - pokazivač, znak koji se mijenja
- kretanje redovima i unutar reda:
  - h, strelica lijevo, ctrl-h - lijevo
  - j, strelica dolje, ctrl-n - dolje
  - k, strelica gore, ctrl-p - gore
  - l, strelica desno, space - desno
- moguće ponoviti kretanje prefiksiranjem (vrijedi za sve naredbe!):
  - primjer: 6h

Pomicanje kursora (2)

- pomicanje unutar ekrana:
  - 0 - apsolutni početak linije
  - ^ - prvi znak u liniji koji nije prazan
  - $ - kraj linije
  - w - početak riječi
  - b - početak prethodne riječi
  -) - početak iduće rečenice
  - (- početak prethodne rečenice

Pomicanje kursora (3)

- pomicanje ekrana:
  - ctrl-U - gore
  - ctrl-D - dolje
  - ctrl-F - naprijed jedan ekran:
  - ctrl-B nazad jedan ekran:
  - H - na prvi znak u datoteci
  - G - na zadnju liniju u datoteci
  - nG - na n-tu liniju u datoteci

Pronalaženje i zamjena teksta

- kako pretražiti za određenim uzorkom (može biti regularni izraz):
  - /uzorak - postavlja kursor na iduću liniju koja sadrži uzorak
  - ?uzorak - postavlja kursor na prethodnu liniju koja sadrži uzorak
  - n - ponavlja pretragu u istom smjeru
  - N - ponavlja pretragu ali u suprotnom smjeru
- zamjena teksta:
  - %s/uzorak/novisadrzaj/g

Ubacivanje teksta

- dotične naredbe automatski postavljaju vi u tekstualni način, a neke i pomiču kursor:
  - a - dodavanje teksta desno od kursora
  - A - dodavanje teksta na kraj reda
  - i - dodavanje ispred kursora
  - o - dodavanje u novom i praznom redu ispod trenutnog
  - O - dodavanje u novi i prazni redak iznad trenutnog

Mijenjanje teksta

- rade isključivo u komandnom načinu rada, ima ih obilje:
  - cw - promijeni trenutnu riječ (vrijedi bilo gdje u riječi)
  - cb - prethodnu riječ
  - cc - cijeli redak
  - dw - obiši trenutnu riječ
  - dd - obriši trenutni redak
  - r - zamijeni trenutni znak sa novim (očekuje novi)
  - R - uključi prepisivanje znakova novima

Mijenjanje teksta (2)

- S - obriši liniju i unesi novi tekst
- x - obriši trenutni znak
- J - spoji dvije linije zajedno
- u - odbaci zadnju promjenu
- ~ - promijeni veliko slovo u malo ili obrnuto

- naredbe promjene (c) označuju područje koje se mijenja sa znakom $

Kopiranje i pomicanje teksta

- brisanje teksta ga sprema u privremeni (nevidljivi) spremnik
- najčešća primjena - obriši, pomakni se, vrati
- naredbe:
  - dw, dd - obriši riječ ili redak (automatski kopira)
  - yw, yy - kopiraj riječ ili redak u spremnik
  - p - vrati spremnik iza kursora
  - P - vrati spremnik prije kursora

Odbacivanje, ponavljanje i spremanje

- odbacivanje i ponavljanje:
  - u - odbacuje zadnju promjenu
  - U - odbacuje sve promjene u retku, ako se kursor nije micao
  - . - ponavlja zadnju akciju ili promjenu
- spremanje:
  - w - sprema sadržaj
  - w! - briše preko postojećeg
  - wq - sprema i izlazi

Otvaranje datoteka

- otvaranje:
  - e datoteka - otvara novu datoteku
  - e! - ponovno otvara tekuću datoteku, zaboravlja promjene
  - r datoteka - dodaje sadržaj datoteke u tekući spremnik
  - r! naredba - izvršava naredbu i dodaje sadržaj u tekući spremnik
  - vi -r datoteka - vraća izgubljeni sadržaj

Podešavanje i postavke

- naredba set:
  - postavlja mogućnosti
  - primjer: `:set all` - ispisuje sve parametre

- konfiguracija - datoteka /etc/.exrc
  - primjer: :set ignorecase
  - primjer: :set wrapmargin=10

Radni zadatak

- pozicionirajte se u vlastiti korisnički direktorij
- skopirajte datoteku /etc/passwd u datoteku sa proizvoljnim imenom
- isprobajte otvaranje, brisanje linija, kopiranje i vraćanje linija
- isprobajte pisanje proizvoljnog teksta i brisanje teksta
- isprobajte pretragu i zamjenu novim izrazom

Dio V:
Preusmjeravanje, filtriranje, ulančavanje

Preusmjeravanje ulaza i izlaza

- redirekcija
  - izlaz ili ulaz preusmjereni na neki drugi resurs osim konzole i tipkovnice
  - korisno za automatiziranje, spremanje rezultata, grešaka, itd
- konvencije (C i ljuska):
  - 0 - standardni ulaz
  - 1 - standardni izlaz
  - 2 - standardni izlaz za greške

Preusmjeravanje ulaza i izlaza (2)

- operatori redirekcije:
  - > - preusmjerava standardni izlaz u neku datoteku
  - >> - preusmjerava standardni izlaz u neku datoteku, dodajući sadržaj na postojeći
  - < - preusmj. standardnog ulaza (iz neke datoteke)
  - << - preusmjeravanje ulaza iz argumenata
  - 1> - isto kao i >, dakle preusmjerenje std. izlaza
  - 2> - preusmjerava standardne greške
  - 2>&1 - preusmjerava std. greške na standardni izlaz, gdje god to bilo

Operator >

- karakteristike:
  - stvara datoteku ako ne postoji
  - generira grešku ako datoteka postoji, ali nema odgovarajuće dozvole za pisanje
  - otvara (i briše) datoteku za pisanje prije izvršavanja naredbe
- primjer: `cat dat > novadat`
- primjer: `ls > lista`
- primjer: `date > datum 2> datum-greska`
- primjer: `date 1> datum 2>&1`

Operator >>

- karakteristike:
  - preusmjerava standardni izlaz i nadodaje na postojeći sadržaj
  - nije destruktivan!
- primjer: `ls -d >> lista`
- primjer: `ls -d /etc >> lista`
- primjer: `cat prva druga >> treca`

Operator <

- karakteristike:
  - ulaz nekog toka ili datoteke se preusmjerava u naredbu
  - najčešće se koristi da se interaktivne programe pretvori u neinteraktivne (izvršavaju se bez korisnikovog uplitanja)
- primjer: `cat < datoteka`
- primjer: `csh < datoteka 1> izlaz 2> greske`

Operator <<

- sintaksa:
  - `naredba << EOF ... EOF`
- karakteristike:
  - omogućava unos sadržaja direktno u naredbu koristeći argumente kao standardni ulaz
  - kraj ulaza je proizvoljan, konvencija EOF
- primjer:
  -
```
cat > datoteka << EOF
ovo ono
EOF
```

Filtri

- filter - naredba koja čita std. ulaz, obrađuje ga i daje neki rezultat na std. izlazu
- primjer - naredba wc:
  - koristi se za brojanje linija, riječi i znakova u ulazu
  - primjer: `wc -l /etc/passwd`
  - parametri:
    - ništa - ispisuje broj linija, riječi i znakova u ulazu
    - -l - ispisuje samo broj linija
    - -w - samo broj riječi
    - -c - samo broj znakova

Sortiranje

- naredba sort:
  - ulaz (ili više ulaza) tretira kao niz znakova (ili brojeva, itd) i sortira ih odgovarajuće u jedinstveni izlaz
  - parametri:
    - -f - sortira bez obzira jesu li velika ili mala slova (mala pretvara interno u velika)
    - -r - sortira opadajuće
    - -u - sortira tako da izbacuje duplikate
    - -m - spaja datoteke kao da su već sortirane
    - -o dat - sortira i sprema rezultat u datoteku dat

Sortiranje (2)

- dodatni parametri:
  - +rbr - sortira počevši od rbr riječi po redu
  - -rbr - zabrana sortiranja od rbr riječi
  - -n - sortira tretirajući ulaz kao ASCII zapisane brojeve
- primjer: `sort -u prva druga`
- primjer: `sort -o prva prva`
- primjer: `sort +1 ulaz`

Radni zadatak

- budite u vašem korisničkom direktoriju
- koristeći isključivo jednu naredbu, sortirajte datoteku /etc/passwd i spremite rezultat u datoteku sortirano
- ispišite koliko linija ima datoteka sortirano

Pretraživanje

- naredba grep:
  - pretražuje ulaz u potrazi za regularnim izrazom u svakom pojedinom retku individualno
  - parametri:
    - -c - prebrojat će broj linija u kojima se pojavljuje izraz
    - -v - logički će negirati će pretragu - dakle traže se svi retci koji ne odgovaraju izrazu
    - -l - ispisuje imena datoteka u kojima je nađen jedan ili više odgovarajućih redaka
    - -n - ispisuje cijeli redak i redni broj retka koji odgovara
    - -i - ignorira razlike između velikih i malih slova
    - -E - omogućava proširene regularne izraze

Pretraživanje (2)

- kratki uvod u regularne izraze:
  - vrlo efikasni - tvore automat sa konačnim brojem stanja
  - ^ - početak reda
  - $ - kraj reda
  - ^niz - zabranjuje pojavljivanje nekog znaka iz niza
  - [prvi-zadnji] - tvori niz znakova redom od prvog do zadnjeg, alfabetski
  - [prvidrugi] - tvori niz znakova od pobrojanih znakova

Pretraživanje (3)

- . - zamjenjuje bilo koji znak
- * - modifikator koji označava da se izraz ispred njega može desiti 0 ili više puta
- + - modifikator koji označava da se izraz ispred njega mora desiti 1 ili više puta (prošireni regex)
- ? - modifikator ... da se izraz može i ne mora pojaviti (prošireni regex)
- | - logičko ili među izrazima (prošireni regex)
- \ - omogućava da se specijalni znakovi tretiraju kao znakovi, odnosno da izgube specijalno značenje ([,], ., *, $, ?, |, ^, \)

Pretraživanje (4)

- primjeri:
  - `grep -v '^9$' /etc/passwd`
  - `grep '[^9]$' /etc/passwd`
  - `grep oot /etc/passwd`
  - `grep '^ro' /etc/passwd`
  - `grep '^[rdbn]' /etc/passwd`
  - `grep -E '^.+:.+:0:.+$' /etc/passwd`
  - itd.

Ulančavanje

- osnovna ideja:
  - standardni izlaz jednog programa postaje standardni ulaz drugom programu
  - operator |
  - najčešće se koriste različiti filteri
  - razmjenjuje se sadržaj među programima bez potrebe za privremenim datotekama
  - primjer: `ls -al $HOME | sort | grep '\.'`
  - primjer: `find . -type f | xargs file`

Ulančavanje (2)

- naredba tee:
  - najčešće se koristi za dupliciranje izlaza
  - što dobije na std. ulazu piše i u datoteku i na vlastiti std. izlaz
  - parametar -a - dodaje na datoteku, umjesto inicijalnog brisanja
  - primjer: `ls -al | tee izlaz | wc -l; cat izlaz`

Ulančavanje (3)

- naredba xargs:
  - stvara naredbu koristeći osnovni naredbeni niz znakova i argumente sa standardnog ulaza
  - za izvršavanje iste naredbe na nizu datoteka
  - parametar -x: brine se o optimalnom popunjavanju linije (maksimalni broj argumenata!)
  - parametar -i: niz u kojem će se zamijeniti svaki {} sa jednim po jednim argumentom iz ulaznog niza
  - primjer: `xargs -x lint -a < cfiles`
  - primjer: `ls | xargs -t -i mv {} {}.old`

Grupiranje

- operator ;:
  - odvaja naredbe
  - one međusobno ne ovise
  - primjer: `ls; cd pero; ls`

- operatori ||:
  - logički ili - ako se prva naredba ne izvrši uspješno, izvršava se druga, dovoljno je da je jedna uspješna
  - primjer: `cat /etc/nema || echo nema`

Grupiranje (2)

- operator &&:
  - logički i
  - && - moraju obje naredbe uspješno završiti, ako prva biva neuspješna, druga se nikad ne izvršava
  - primjer: `ls /etc/ima && echo ima`
  - primjer: `ls /etc/passwd && echo ima`
- operatori ():
  - spaja se u grupu i tretira kao jedna naredba
  - primjer: `(cd /etc; ls passwd) > rezultat`

Grupiranje (3)

- glavni činitelj u grupiranju - izlazni status naredbe ili programa
  - u ljusci je to varijabla $?
  - primjer: `print $?`
  - omogućava da se naredbe izvrše u ovisnosti o uspjehu ili neuspjehu
  - svi programi bi standardno trebali vraćati operacijskom sustavu i/ili ljusci kakvu vrijednost

Radni zadatak

- pozicionirajte se u vaš tekući direktorij
- ne koristeći privremene datoteke, ispišite sve datoteke u /etc direktoriju, sortirajte ispis i iz takvog ispisa spremite u datoteku privremeni samo one retke koji sadrže riječ new

Awk

- vrlo kompleksni program - praktički programski jezik nalik na C
- najčešće se koristi za složenije obrade više teksta, uspješno zamjenjuje većinu standardnih filtera i alata
- prolaženje po tekstu i pretraživanje nije uvijek ograničeno na linije, već može i kroz grupe linija

Awk (2)

- zapis - najčešće linija teksta
- polje
  - dijelovi zapisa, standardno su to riječi
  - standardni separator je razmak
  - prebrojavanje počinje od 1 (ne od 0!) za prvo polje
  - 0-to polje je cijeli zapis (redak)
  - $0, $1, $2, itd. su standardno definirane varijable odnosno polja

Awk - programiranje

- struktura programa:
  - uzorak { akcija } uzorak2 { akcija2 } ...
- može se pozivati iz komandne linije, a može i interpretirati odgovarajući program
- izrazi:
  - BEGIN akcija - akcija se izvršava samo jednom i to prije ikakve obrade teksta (prolog)
  - END akcija - akcija se izvršava samo jednom i to poslije svih obrada (epilog)

Awk - programiranje (2)

  - NR - broj pročitanih zapisa tijekom obrade
  - NF - broj polja u tekućem zapisu
  - FS - separator ulaznih polja
  - OFS - separator izlaznih polja
- operatori usporedbe:
  - == - jednak je
  - != - nije jednak
  - > - veći od
  - < - manji od

Awk - programiranje (3)

  - >= - veći i jednak
  - <= - manji i jednak
- logički operatori:
  - and, && - logičko i
  - or, || - logičko ili

- primjer: `'BEGIN { FS=":"; OFS="\t" } { print $1, $5 }'`

Awk - programiranje (4)

- primjer:
  - BEGIN { FS=":" }
  - { if (($3 >= 50) && ($3 <= 99)) arr[$3] = 1}
    END {
    for (j = 50; j <= 99; ++j)
    if (!(j in arr)) { print j; break }
    }

Dio VI:
Kontrola procesa

Uvod

- proces - program koji se izvršava
- operacijski sustav - kontrolira procese, raspoređuje im resurse (procesorsko vrijeme, memorija, itd)
- osobine:
  - može biti stvoren i ubijen
  - zauzima resurse
  - može stvoriti druge procese
  - može komunicirati sa ostalim procesima

Uvod (2)

  - ima vlastitu okolinu:
    - nasljeđena je od procesa roditelja
    - sastoji se od informacija potrebnih procesu
    - može biti promijenjena i od procesa i od ljuske
- dotična okolina sadržava:
  - PID i PGID (ID procesne grupe)
  - otvorene datoteke
  - radni direktorij
  - masku za dozvole
  - realne i efektivne UID i GID

Uvod (3)

  - limite na sistemske resurse
    - maksimalnu veličinu datoteka
    - maksimalnu količinu memorije
    - itd.
  - akcije koje se obavljaju po dospjeću signala
  - set imenovanih varijabli
- program init:
  - prvi proces kojeg pokreće kernel, PID 1
  - iz njega direktno ili indirektno nastaju svi ostali procesi, uključujući i korisničke ljuske i slično

Poslovi u ljusci

- tekući posao (foreground)
  - svaka naredba, svaki program se standardno izvršava interaktivno i na trenutnom terminalu
  - ljuska ne prima ulaz sve dok program ne završi
  - standardno ponašanje
- poslovi u pozadini (background)
  - procese je moguće poslati i u pozadinsku obradu
  - moguće je imati više procesa u pozadini

Poslovi u ljusci (2)

- takvi procesi nemaju direktnu kontrolu nad ulazom i izlazom ljuske
- ljuska normalno prihvaća daljnju komunikaciju
- moguće koristiti za sve neinteraktivne procese (filteri, proračuni, itd)
- sintaksa: program &
- simbol & informira ljusku o pokretanju u pozadini
- treba pripaziti da programi u pozadini imaju ulaz i izlaz definiran kroz preusmjeravanja iz datoteka, a ne konzole

Ispis procesa

- naredba ps:
  - ispisuje procese i informacije o njima
  - tipovi informacija:
    - PID - jedinstveni broj procesa
    - TTY - oznaka terminala na kojem je proces aktivan
    - S - stanje procesa, do 4 slova ga označuju
    - TIME - akumulirano vrijeme procesa
    - CMD - ime procesa i njeni argumenti
  - argumenti naredbi:
    - -a - informacije o svim procesima osim onima bez terminala i onima koji nisu vlasnik grupe

Ispis procesa (2)

- -e - informacije o svim procesima
- -f - dodatne informacije (PPID, CPU aktivnost u %, vrijeme kad je proces pokrenut)
- -l - još dodatnih informacija (zastavice, UID, PPID, %CPU, prioritet, interni raspored procesa, veličina procesa, klasifikacija statusa procesa)
- -u lista - ispisuje procese korisnika iz liste

– primjer: `ps -fu $USER`
– primjer: `ps xuaw | more`
– primjer: `ps -elf | more`

Ispis poslova

- naredba jobs:
  - ispisuje pozadinske poslove ljuske
  - ispisuje broj posla, koji se koristi u naredbama bg i fg
  - argumenti:
    - -l - ispisuje i imena i PID
    - -p - ispisuje samo PID
  - primjer: `jobs -l`

Upravljanje poslovima

- suspendiranje posla:
  - slanje signala aktivnom/tekućem posla za privremeni prekid rada
  - proces trenutačno prekida izvođenje, ostajući u smrznutom stanju
  - najčešće ctrl-z kombinacija tipki
- naredba bg:
  - slanje suspendiranog posla u pozadinski rad
  - proces normalno nastavlja raditi
  - kao argument prima broj posla (naredba jobs)

Upravljanje poslovima (2)

- naredba fg:
  - pozadinski ili trenutno suspendiran posao vraća u aktivni/tekući rad, nazad na interaktivnu konzolu
  - proces postaje standardni interaktivni na konzoli
  - moguće opet poslati u pozadinu, itd.
  - proces normalno nastavlja raditi
  - kao argument prima broj posla (naredba jobs)
- primjeri:
  - `bg; bg %3; bg 2344`
  - `fg; fg %1; fg 32432`

Radni zadatak

- pokrenite vi editor bez imena datoteke
- upišite par znakova teksta
- suspendirajte ga
- pogledajte vlastitu listu procesa
- pogledajte sve procese na sustavu
- vratite proces nazad u interaktivni rad
- izađite iz editora bez snimanja

Signali i procesi

- posao je vezan uz ljusku
- općenito, proces ne mora biti vezan uz ljusku
- naredba kill:
  - šalje signale procesima
  - ovisno o signalu, proces ili ljuska ili jezgra obavljaju predodređene akcije nad procesom
  - sintaksa: kill [-signal] proces
  - moguće je proces terminirati, vratiti u pozadinu, suspendirati, itd.

Signali i procesi (2)

- važniji signali:
  - 1, HUP - pročitaj ponovo konfiguraciju
  - 2, INT - prekini to što radiš
  - 3, QUIT - prekini sa izvršavanjem
  - 9, KILL - terminiraj odmah proces bez odgode
  - 15, TERM - prekini sa izvršavanjem
  - 20, CHLD - dijete je završilo sa radom
  - 11, SEGV - nedozvoljeni pristup memoriji
  - primjer: `kill -9 21343; kill -1 %1`
  - primjer: `kill -9 -1`

Prioriteti procesa

- prioritet procesa:
  - operacijski sustav ih koristi interno
  - koliko se često/brzo program izvršava
  - raspon od -20 (najbrži, obično kernel procesi) do +19 (najsporiji)
  - standardno se podrazumijeva +10
  - jedino administrator može koristiti negativne
- naredba nice:
  - sintaksa: `nice [-n prioritet] naredba [argumenti]`

Prioriteti procesa (2)

  - primjer: `nice -n 15 find / -name passwd -print`
- standardni prioriteti:
  - +19 - izvršava se kad ništa ne radi drugo na sustavu
  - +10 - standardna vrijednost za nice naredbu bez parametra n
  - 0 - standardna vrijednost procesa
  - -1 do -20 - iznimno brzi rad, gušenje ostatka sustava, nije preporučljivo

Vremensko upravljanje

- procese je moguće pokrenuti u budućnosti:
  - jednom - koristeći naredbe at i batch
  - redovno i višestruko - naredba crontab
- naredba batch:
  - sintaksa - upisuju se naredbe u std ulaz
  - izvršit će se naredbe kad to dopusti opterećenje sustava
- naredba at:
  - omogućava definiranje točnog vremena za izvršenje

Vremensko upravljanje (2)

- parametri:
  - -c - csh
  - -k - ksh
  - -s - sh
  - -m - šalje e-mail kad i ako uspješno završi
  - -l - ispisuje poslove u čekanju
  - -r broj - otkazuje posao u čekanju, bilo onaj od batch bilo od at naredbe
  - -r -u korisnik - otkazuje sve poslove za dotičnog korisnika

Vremensko upravljanje (3)

- u praksi se rijetko koristi - daleko češće se koristi cron
- primjeri:
  - `at -km now +2 days nesto.ksh`
  - `batch ...`

- crond - servis koji je uvijek aktivan
  - izvršava periodičke poslove, sprema izvještaje

Vremensko upravljanje (4)

- procesi automatski pozvani:
  - izvršavaju se u korisnikovoj ljusci
  - dobivaju standardno okruženje ($HOME, $LOGNAME, $SHELL, itd)
- naredba crontab:
  - za upravljanje konfiguracijom
  - argumenti:
    - -l - ispis i -v za detaljni ispis
    - -e - editiranje
    - -r - brisanje

Vremensko upravljanje (5)

- standardni crontab zapis:
  - niz redaka iste sintakse
  - `mm hh DD MM WW naredba`
    - mm - minuta, hh - sat, DD - dan, MM - mjesec
    - WW - dan u tjednu (0 - 6, 0 je nedjelja)
  - svako polje moguće zamijeniti sa *
  - moguć je i raspon sa oblikom prvi-drugi
  - moguće je i napisati listu sa , kao separatorom
  - primjer: `0 * * * * nice dailystrips ...`
  - primjer: `0 3 * * * nice rotirajlogove`

Radni zadatak

- napišite cron skriptu koja će svaki dan u ponoć pokrenuti pretraživanje /etc direktorija, spremiti rezultat u vaš korisnički direktorij sa imenom rezultat
- pretraživanje mora biti pod prioritetom 10

Kraj i diskusija