

## Sietove sluzby

### Runlevel

/etc/init.d/ - spustanie a zastavovanie sluzieb  
/etc/rc0.d/ - halt  
/etc/rc1.d/ 1 ... 5  
/etc/rc5.d/ - samotne spustanie pre jednotlivé runlevely single-userov  
/etc/rc6.d/ - root

/etc/inetd.conf - demon pre sietove sluzby  
/etc/services - rozpis sietovych sluzieb a portov  
ftp,ssh,telnet,smtp,dns,finger,http,pop3,samba,imap,https,ipp,  
talk,imaps,pop3s,

### TCP - wrapper

samotny wrapper je /usr/sbin/tcpd skontroluje ip podľa pravidiel v host.allow/danied a postúpi ďalej ...  
pre ftp zmeniť v /etc/inetd.conf:

**ftp stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.ftpd**

a zdefinovať v /etc/hosts.allow" a "/etc/hosts.deny pravidla

**ALL: .example.com EXCEPT server.example.com**

pre testovanie demonov **tcpdmatch in.ftpd 192.168.1.7**

### Sledovanie systemu

netstat - sledovanie siete a prepojení , rozložení portov *netstat -pevan*

logcheck - získanie a filtrovanie logov servera  
- konfigurácia: /etc/logcheck/logcheck.conf

ippl - slúži na logovanie pokusov o prístupy na porty vášho servera pomocou protokolov  
TCP, UDP a ICMP.  
- konfigurácia /etc/ippl.conf

## POSTFIX

Moduly z ktorých sa skladá

- **pickup**: čaká na lokálne napísané maily a presmeruje ich obsah na démona "cleanup"
- **cleanup**: spracúva prijatý mail (pridáva chýbajúce hlavičky a pod.), vkladá ho do frontu prijatých mailov a informuje démona "qmgr" o jeho príchode
- **qmgr**: čaká na prijaté maily a zabezpečuje ich doručenie. Spôsob doručenia určí démon "trivial-rewrite"
- **trivial-rewrite**: prepisuje adresu do štandardizovanej formy. Démon pripája meno domény k lokálnym mailom bez jej uvedenia a pod. Okrem toho určuje, čo sa stane s mailom, ako a kam sa bude doručovať na základe adresy
- **local**: doručuje mail do lokálnych schránok na serveri
- **smtp**: smtp klient Postfixu. Doručuje maily z mailového frontu, ktoré sú určené pre iné mailové servery.

Adresare Postfixu:

/etc/postfix: adresár s konfiguračnými súbormi  
/etc/postfix/master.cf: hlavný konfiguračný súbor služieb, nastavenia jednotlivých služieb  
/etc/postfix/main.cf: hlavný konfiguračný súbor "postfixu" s nastaveniami doručovania a odosielania pošty  
/etc/aliases: súbor s mailovými aliasmi používateľov platnými pre lokálny počítač  
/etc/postfix/\*: všetky ostatné súbory v tomto adresári sú tabuľky "postfixu" v textovom a databázovom tvare (db, dbm)  
/var/spool/postfix/: hlavný adresár s dátovými štruktúrami "postfixu" počas jeho prevádzky.  
Sú tu uložené všetky práve doručované, prijímané a nedoručiteľné maily. Tento adresár slúži ako základný adresár pre jednotlivé služby - tie využívajú jeho podadresáre.

S Postfixom môžete vykonávať tieto bežné operácie:

- Spustenie démona: `"/etc/init.d/postfix start"`
  - Zastavenie démona: `"/etc/init.d/postfix stop"`
  - Reštartovanie démona: `"/etc/init.d/postfix restart"`
  - Opätovné nahranie konfigurácie démona (po zmenách v konfigurácii): `"/etc/init.d/postfix reload"`
  - Kontrola konfiguračných súborov: `"postfix check"`
- (Pokus o) doručenie všetkých čakajúcich neodoslaných mailov (napríklad po odstránení nejakej chyby na vašom serveri/sieti):  
`"postfix flush"`
- Aktualizácia aliasovej databázy (`"/etc/aliases", "/etc/postfix/aliases.*"`): `"newaliases"`
  - Aktualizácia inej databázy (`"/etc/postfix/*", "/etc/postfix/*.db"`): `"cd /etc/postfix ; postmap meno_suboru"`
  - Výpis stavu mailového frontu: `"mailq"`

## Firewall

Firewall je sieťové zariadenie a/alebo softvér, ktorého úlohou je oddeliť siete s rôznymi prístupovými právami (typicky napr. Internet a Intranet) a kontrolovať tok dát medzi týmito sieťami.

Kontrola údajov prebieha na základe aplikovania **pravidiel**, ktoré určujú **podmienky** a **akcie**. Podmienky sa stanovujú pre údaje, ktoré možno získať z dátového toku (napr. zdrojová, cieľová adresa, zdrojový alebo cieľový port a rôzne iné). Úlohou firewallu je vyhodnotiť podmienky a ak je podmienka splnená, vykoná sa akcia. Dve základné akcie sú "povoliť dátový tok" a "zamietnuť dátový tok". Po vykonaní takejto akcie firewall prestane paket spracovávať. Existujú však aj iné akcie, ktoré neurčujú osud paketu a slúžia napr. na logovanie hlavičiek paketu, zmenu hlavičiek paketu a podobne.

Ďalšou vlastnosťou firewallu, ktorá sa často používa, i keď nejde o filtrovanie, je schopnosť prekladu adres (Network Address Translation - NAT). NAT umožňuje zmeniť zdrojové a cieľové adresy v paketoch, čím sa najčastejšie umožňuje komunikácia so sieťami s privátnymi adresami (napr. 10.0.0.0/8). Aj preklad adres prebieha pomocou pravidiel.

```
iptables          - sprava paketoveho filtra
iptables -nvL     - vypis nastavenia
iptables -A-I-D-F INPUT|OUTPUT|FORWARD|  ACCEPT|DROP|QUEUE|RETURN|
```

**iptables -F input|output|forward|prerouting|postrouting**

**iptables -P INPUT DROP** - zahadzovanie paketov smerovane na firewall

**iptables -P FORWARD DROP** – zahadzovanie prechadzajucich paketov cez firewall

**iptables -P OUTPUT ACCEPT** – povolenie odchadzajucich paketov z firewalu

**iptables -A INPUT -p tcp -j DROP** – vsetky prichadzajuce pakety TCP budu zahodene

**iptables -A INPUT -p udp -s 10.6.6.0/255.255.255.0 -j DROP** – vsetky tcp v rozsahu adres budu zahodene

**iptables -A FORWARD -p udp -s ! 10.0.0.1 -j DROP** – ktore neprechadzaju z rozsaku ip

**iptables -A FORWARD -s 192.168.0.0/16 -d 10.0.0.1 -j DROP** - všetky pakety zo siete 192.168.0.0/16 smerujúce na 10.0.0.1 cez tento počítač budú zahodené.

**iptables -A INPUT -p tcp -s ! 10.0.0.0/8 --dport pop3 -j DROP** - všetky TCP pakety, ktoré neprichádzajú zo siete 10.0.0.0/8 na port "pop3" budú zahodené

**iptables -A INPUT -p tcp -s 10.0.0.0/8 -d 10.0.0.1 --dport ssh -j ACCEPT**

**iptables -A INPUT -p tcp --syn -j DROP** Prvé povolí prichádzajúce pakety zo siete 10.0.0.0/8 na adresu 10.0.0.1 (predpokladá sa, že je to vnútorná adresa firewallu) na cieľový port "ssh". Druhé pravidlo zahodí všetky pakety, ktoré sa pokúšajú vytvoriť spojenie na počítač, na ktorom beží firewall

**iptables -A FORWARD -p udp -s 192.168.0.1 -d 10.0.0.0/8 -j ACCEPT**

**iptables -A FORWARD -p udp -d 10.0.0.0/8 -j LOG --log-prefix test.log**

**iptables -A FORWARD -p udp -d 10.0.0.0/8 -j DROP** - kombinácia pravidiel. Prvé pravidlo povolí UDP pakety z adresy 192.168.0.1 smerujúce do siete 10.0.0.0/8. Tretie pravidlo zakáže všetky (ostatné) UDP pakety smerujúce do siete 10.0.0.0/8. Všimnite si, ako záleží na poradí! Ak vymeníte poradie pravidiel, UDP pakety z adresy 192.168.0.1 sa nikdy do siete 10.0.0.0/8 nedostanú! Druhé pravidlo loguje všetky pakety, ktoré neprejdú tretím pravidlom

## **PROXI SERVER**

Proxy server je špeciálny typ servera - prostredníka v komunikácii, ktorý sa umiestňuje medzi klienta a servery, s ktorými komunikuje. Proxy server sa tvári voči klientovi ako server a voči serveru ako klient. Výhodou je, že proxy server pozná požiadavku klienta a vie mu doručiť odpoveď, aj keď klient samotný nemôže alebo nevie (kvôli obmedzeniam alebo parametrom siete) priamo komunikovať so vzdialeným serverom.

SQUID – opensourcovy proxy server., spustenie */etc/init.d/squid start|stop|reload|status konfiguracia je v /etc/squid.conf*

```
#Recommended minimum configuration:
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl SSL_ports port 443 563
acl Safe_ports port 80          # http
acl Safe_ports port 21         # ftp
acl Safe_ports port 443 563    # https, snews
acl Safe_ports port 70         # gopher
acl Safe_ports port 210       # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280       # http-mgmt
acl Safe_ports port 488       # gss-http
acl Safe_ports port 591       # filemaker
acl Safe_ports port 777       # multiling http
acl CONNECT method CONNECT
```

## **FTP**

instalacia vsftpd , /etc/init.d/vsftpd start

## **BASH sript**

### **FOR**

```
for (( a=1 ; $a-4 ; a=$a+1 ))do echo $a; done;
for a in A B C D do echo $a; done;
for a in $( seq 9 )do wget http://www.masinky.cz/obr${a}.jpg; done;
for a in soubor*.txt ; do rm -f $a ; done;
```

### **IF**

```
if grep "/bin/bash$" /etc/passwd > /dev/null ; then
echo "Bash majú nastaveni:"
awk -F ":" , $7 == "/bin/bash" {print $1}` /etc/passwd
else
echo "Nikdo :-( "
fi
```

### **WHILE**

```
while true; do
echo "StisknĚte klávesu"
read -n 1
echo -n "PrávĚ je "; date
done
```

```

CASE
echo "Určení osoby zájmena"
echo -n "Zadejte osobní zájmeno: "
read zajmeno
case $zajmeno in
  já|my) echo "1. os." ;;
  [vt]y) echo "2. os." ;;
  on|on[aioy]) echo "3. os." ;;
  *) echo "Chyba" ;;
esac

```

### **SELECT**

```

echo "Zvolte typ oken:"
select okno in dřevěná plastová ikspé; do
echo "Nejděravější okna jsou "$okno
break
done

```

### **POZICNE PARAMETRE**

```

#!/bin/bash
if [ "$1" = "-h" ]; then
  echo "pomoc pro uživatele!"
else
  echo "uživatel nechce pomoc!"
fi

```

### **EXPANZIA PARAMETRU**

**\${parametr:=slovo}** - vypíše hodnotu parametru a, pokud je tento nulová (nenastavený), zobrazí místo něj zadaný řetězec (slovo). Přitom ale zároveň změní hodnotu parametru tak, že do něj dané slovo vloží.

**\${parametr:?slovo}** — další obdoba první varianty s tím, že slovo se vypíše v případě nulové hodnoty parametru, ne však jako běžné, nýbrž chybový výstup. Pokud to nastane v neinteraktivním shellu, bude shell (i skript) ukončen.

**\${parametr:+slovo}** — Pokud je parametr nulový nebo nenastavený, nic se nestane. Pokud však nějakou hodnotu obsahuje, zobrazí se slovo.

**\${parametr:offset}** — Bash v takovém případě uřízne zadaný počet znaků ze začátku vypisovaného řetězce. Např. echo \${SHELL} vypíše /bin/bash, zatímco echo \${SHELL:5} jen bash, protože prvních 5 znaků ubral.

**\${parametr:offset:delka}** — pokud vám nestačí řezat pouze od začátku, můžete přidat další dvojtečku a za ni počet znaků, které chcete od offsetem zadaného místa zobrazit (nikoli uřezat zezadu). echo \${SHELL:5:2} potom ukáže jen "ba", tedy šestý a sedmý znak hodnoty \$SHELL.

**\${parametr/vyraz/retezec}** — a to nejlepší nakonec. Bash umí v řetězcích nahrazovat určený výraz zadaným řetězcem. Řekněme, že chceme při zobrazení zvýraznit všechny výskyty bin v proměnné PATH velkými písmeny.

## ARITMETICKA EXPANZE

- **\$(a++)**, resp. **\$(a--)** — vypíše hodnotu proměnné a a následně ji zvýší (resp. sníží) o jedničku.
- **\$(++a)**, resp. **\$(--a)** — nejprve zvýší (resp. sníží) hodnotu proměnné a o jedničku a potom tuto novou hodnotu vypíše.
- **\$(2\*\*3)** — třetí mocnina dvou, tj. 2 na třetí (8).
- **\$(7%2)** — výsledkem je zbytek po dělení 7/2. Jak jsem již zmínil, Bash umí pracovat jen s celými čísly, proto jako výsledek dělení 7/2 vrátí 3, zbytek (7%2) bude 1.
- **\$(a==5)** — vyhodnotí, zda se hodnota proměnné a rovná číslu 5. Výsledkem tohoto výrazu je pravda (1) nebo nepravda (0). Obdobně funguje vyhodnocování nerovností (!=, <=, >=, <, >).
- **\$(4<1)** — provede bitový posun doleva, tedy 3. bit (hodnota 4) o 1, tj. na 4. bit (hodnota 8) a výsledkem bude 8.
- **\$(4>1)** — provede bitový posun doprava, tedy 3. bit (hodnota 4) o 1, tj. na 2. bit (hodnota 2) a výsledek je 2.
- **\$(5&6)** — bitové AND, vypíše hodnotu společných bitů, v našem případě 5 (1. a 3. bit) a 6 (2. a 3. bit) je společný 3. bit, tedy hodnota 4.
- **\$(5|6)** — bitové OR, vypíše hodnotu bitů, které se vyskytují alespoň na jedné straně, v tomto případě 5 (1. a 3. bit) a 6 (2. a 3. bit) jsou použity bity 1, 2 a 3, tedy hodnota 1+2+4=7.
- **\$(5^6)** — bitové exkluzivní OR, vypíše hodnotu bitů, které se vyskytují pouze na jedné nebo pouze na druhé straně, zde tedy 5 (1. a 3. bit) a 6 (2. a 3. bit) má společný bit 3 a exkluzivní zůstávají 1 a 2, dohromady 3.
- **\$(a&b)** — logické AND, nabývá hodnoty 1, pokud jsou hodnoty a i b nenulové, jinak nabývá hodnoty 0.
- **\$(a|b)** — logické OR, nabývá hodnoty 1, pokud alespoň jedna z hodnot a a b je nenulová, jinak nabývá hodnoty 0.

## MYSQL

```
mysqldump nazov_databazy nazov_tabulky > nazov_suboru.sql - export db do subory
```

```
mysqldump nazov_databazy < nazov_suboru.sql - import zo suboru do db
```

```
LOAD DATA LOCAL INFILE 'udaje.txt' INTO TABLE udaje FIELDS TERMINATED BY ';' ENCLOSED BY ""  
LINES TERMINATED BY '\n'; - import do tabulky zo suboru separator stlpcov je ';' hodnoty zaznamou su medzi  
"'" , riadok je ukonceny \n
```

## CVS – Concurrent version systems

```
cvs import -m "zalozeni projektu" prvniprojekt poc-tym poc-verze - zalozenie projektu  
cvs checkout prvniprojekt - vytvory a naplni adresar src. subormi  
cvs commit -m "nejaka poznamka" jedensubor.txt - zmeni obsah suboru v cvs  
cvs add -m "nejaka poznamka" jmeno_souboru - prida dalsi subor do cvs  
cvs add -kb -m "nejaka poznamka" jmeno_souboru - prida binarny subor do cvs  
cvs update jmeno_souboru - upravu lokalny adresar podľa obsahu v cvs  
cvs commit -m "oprava na binární verzi" jmeno_souboru - zemni obsak v cvs  
cvs remove -f jmeno_souboru - zaze subor z cvs  
cvs log jmeno_souboru > zmeny.log - vypis zmienv suboru cvs do logu  
cvs log -d "1 week ago" - vypis zmienv obdobi ...  
cvs log -d "1999/12/01" - vypis zmienv obdobi ...  
cvs log -d "1999/11/01 <= 1999/11/30" - vypis zmienv obdobi ...
```