

# VSO – Laboratorijske vaje 5




Omrežne nastavitve  
SSH, scp, sftp  
Storitve, NFS


# Omrežne nastavitve

- *Sistem -> Možnosti -> Omrežne povezave*

**Omrežne povezave**

Žično | Brezžično | Mobilni širokopolasovni dostop | VPN | DSL

Ime	Nazadnje uporabljeno	
Auto eth0	Pred 2 minutama	 Dodaj
		 Uredi ...
		 Izbriši ...

 Zapri

**Urejanje Auto eth0**



Ime povezave: Auto eth0

Samodejna povezava

Žično | Varnost 802.1x | Nastavitve IPv4 | Nastavitve IPv6

Način: Samodejno (DHCP)

**Naslovi**


Naslov	Omrežna maska	Prehod	
			 Dodaj
			 Izbriši

Strežniki DNS:


Iskalne domene:


ID odjemalca DHCP:

Zahtevaj, da se konča dodeljevanje naslovov IPv4 za to povezavo

 Smeri ...

Na voljo vsem uporabnikom

 Prekliči

 Shrani ...

# OpenSSH

- Uporabljamo ga za varen (kriptiran) dostop do oddaljenih računalnikov preko ukazne lupine.
- Nameščen moramo imeti paket *openssh-server* in *openssh-client*.

```
ssh -l uporabnik imeracunalnika
```

# scp

- Ukaz *scp* - secure copy - uporabljamo za varno kopiranje datotek iz enega računalnika na drugega.
- na drugem računalniku moramo imeti uporabniško ime in poznati ustrezno geslo.
- *scp pot1 pot2* (podobno kot ukaz *cp*)
- Zgledi:
  - *scp test.txt janez@r3-8.fri.uni-lj.si:~/poddirektorij*
  - *scp -r janez@r3-8.fri.uni-lj.si:/home/janez .*

# sftp

- Deluje podobno kot *ftp*
- Prijavimo se z uporabniškim imenom in geslom
- Interaktivna *ftp* lupina z odzivnikom *sftp>*
- Hkrati smo v trenutnem direktoriju na svojem računalniku in prijavljeni na drugi strani.

# sftp - ukazi

- **Osnovni ukazi na drugi strani**

- *bye* (izhod)
- *mkdir imedirektorija*
- *cd direktorij*
- *get datoteka*  
(kopiranje naš -> drugi)
- *put datoteka*  
(kopiranje drugi -> naš)
- *bin*  
(podatki naj se prenašajo binarno)
- *ls*

- **Ukazi za naš računalnik**

- *!mkdir imedirektorija*
- *!cd direktorij*
- *!ls*

# Prijava s pomočjo RSA ključev

- RSA ključi omogočajo prijavo brez gesel.
- Mi imamo zasebni ključ.
- Na računalnik, na katerega se prijavljamo, moramo namestiti javni ključ.
  
- Oddaljeni računalnik ob poskusu prijave:
  - Izbere neko veliko naključno število.
  - Ga šifrira z našim javnim ključem in pošlje k nam.
  
- Naš računalnik:
  - Ker imamo samo mi zasebni ključ, znamo samo mi dešifrirati število.
  - Dešifrirano število pošljemo nazaj oddaljenemu računalniku.
  
- Oddaljeni računalnik
  - Če je število pravilno, nam oddaljeni računalnik omogoči dostop.

# Ključī

- Zaženemo ukaz
- `ssh-keygen -t rsa`
- Potrdimo predlagan direktorij `~/.ssh/id_rsa`
- Dvakrat vnesemo **geslo za zaščito zasebnega ključa.**
- V `~/.ssh` se generirata dve datoteki `id_rsa.pub`, ki je **javni ključ**, in `id_rsa`, ki je **zasebni ključ**.
- direktorij `~/.ssh` moramo z ukazom `chmod` zaščititi na 700.
- javni ključ s `scp` prekopiramo na drugi računalnik v delovno področje uporabnika na drugem računalniku, pod katerim se prijavljamo.
- Prijavimo se na drugi računalnik in z ukazom  
`cat id_rsa.pub >> ~/.ssh/authorized_keys`  
pripravimo javni ključ datoteki `./ssh/authorized_keys`



# Prijava z RSA ključi

Vsakič, ko se prijavljamo na oddaljeni računalnik (na katerem je nameščen javni ključ) se prijavimo z

```
ssh uporabnik@ime_oddaljenega_racunalnika
```

Vnesti moramo geslo za zaščito zasebnega ključa. Geslo za zaščito zasebnega ključa se na lokalnem računalniku uporabi za dešifriranje zasebnega ključa (če nam ga kdo ukrade, si ne more takoj pomagati z njim).

# Prijava z RSA ključi

- Če nastavimo prazno geslo, moramo še posebej paziti, da nam kdo ne ukrade zasebnega ključa.

# Storitve

- V Linuxu se imenujejo *services*
- Tečejo v ozadju in nudijo neke storitve.
- Vedeti moramo, kaj je njihova naloga

# Storitve in ukazna vrstica

- Stanje
  - */sbin/service ime status*
- Zagon
  - */sbin/service ime start*
- Ustavljanje
  - */sbin/service ime stop*
- Ponoven zagon
  - */sbin/service ime restart*

*Če je /sbin v iskalni poti, ga ni potrebno navajati.  
Prijavljeni moramo biti kot root (su -).*

# NFS

- Network File System
- “Protokol” skupno rabo datotek.
- Na računalniku, na katerem je direktorij dan v skupno rabo, mora biti nameščen NFS strežnik (NFS server).
- Računalnik, ki prilaplja direktorij v skupni, mora imeti NFS odjemalca (NFS client).

# Dajanje map v skupno rabo iz terminala

- Odpremo datoteko `/etc/exports`
- Dodamo vnos, npr.  
`/home/pub 192.168.123.*(ro)`
- Spremembe začnejo veljati po ponovnem zagonu nfs strežnika:  
**#service nfs-kernel-server restart**
- Vsak direktorij v skupni rabi je določen z eno vrstico
- Prvi podatek se začne na začetku vrstice in je pot do direktorija.
- Potem je nekaj presledkov, za njimi pa imena ali IP naslovi računalnikov, ki lahko do direktorija dostopajo (\* pomeni karkoli).
- Tako za IP naslovi (brez presledkov) je oklepaj, v njem pa opcije, ločene z vejicami in brez presledkov.

# Opcije pri dajanju v skupno rabo

- *ro* - samo branje
- *rw* - branje in pisanje
- *no\_root\_squash* - root iz drugega računalnika ima root pravice na direktoriju
- *all\_squash* - vsi uporabniki, ki se priklopijo na direktorij, se prevedejo na anonimnega uporabnika (nfsnobody)
- ...

# Postopek

- Prijavite se kot *root*.
- V */etc/exports* vpišite direktorij, ki ga dajete v skupno rabo (skupaj z ustreznimi nastavitvami)
- Če stvari ne delujejo, ponovno zaženite storitev *nfs*
  - */sbin/service nfs-kernel-server restart*



# Priklop direktorija danega v skupno rabo

- Vedeti moramo, na katerem računalniku in kateri direktorij želimo priklopiti.
- Uporabimo ukaz mount:

```
mount -t nfs ip_naslov:/pot/do/direktorija /lokalni_direktorij
```

- direktorij na drugem računalniku je podan absolutno.
- Pravice dostopa delujejo glede na številke UID in GID (uporabnik, skupina).
- Dober način uporabe - opcija `all_squash` (potem smo predstavljeni kot uporabnik *nobody*).

# Mount in /etc/fstab

- Ukaz mount v vsej splošnosti lahko izvede samo *root*.
- Drugi uporabniki lahko priklopijo direktorij le, če je to navedeno v */etc/fstab*.
- Vrstica v *fstab*:
  - /dev/cdrom /mnt/cdrom udf,iso9660 noauto,owner,kudzu,ro 0 0
  - prevozni3.fri.uni-lj.si:/home/pub nfs user 0 0

# fstab

- Polja ločena s presledki:
  - naprava ali direktorij na drugem računalniku, dan v skupno rabo,
  - lokalni direktorij priklopa,
  - z vejicami ločeni datotečni sistemi (brez presledkov vmes),
  - z vejicami ločene opcije (brez presledkov vmes) - *user,ro,owner,gid=,mode=,noauto...*
  - zadnji dve polji ponavadi postavimo na 0 0 (glej *man fstab*).

# Omogočanje priklopa NFS direktorija uporabniku

- Želimo, da se lahko 'navadni' uporabnik priklopi na direktorij:  
`testni.fri.uni-lj.si:/home/pub`
- in jo priklopi na  
`/mnt/testni`
- *V `/etc/fstab` dodamo:*  
`testni.fri.uni-lj.si:/home/pub /mnt/testni nfs user 0 0`
- *'navadni' uporabnik priklopi direktorij z ukazom*  
`mount /mnt/testni`

# Samba

- Omogoča nam, da dajemo direktorije v skupno rabo, tako da se jih da priklopiti v Linuxu in Windows.
- Za priklop direktorija, danega v skupno rabo iz Windows računalnika na Linux računalnik, potrebujemo na slednjem le Samba odjemalca (client).
- Če želimo dati direktorij v skupno rabo na Linux računalniku, tako da ga lahko priklopimo v Windows sistemu, potrebujemo na Linux računalniku Samba strežnik (server).

# Uporaba Samba odjemalca

- Ukazna vrstica:
  - `smbclient //ime_racunalnika/direktorij_v_sk_rabi -U uporabnik`
- Če smo uporabnik miha in se želimo priklopiti na računalnik *arhimed.fri.uni-lj.si* na direktorij, ki je dan v skupno rabo pod imenom *miha\$* (npr. naš mrežni disk v domeni STD):
  - `smbclient //arhimed.fri.uni-lj.si/miha$ -U miha`
- Uporabljamo ga na enak način kot *ftp*.
- Ukaz *help* za pomoč.

# mount -t smbfs

- Priklop *smb* direktorija, danega v skupno rabo na direktorij.
- Uporaba:
  - *mount -t smbfs //racunalnik/direktorij\_v\_sk\_rabi /nasdirektorij*
  - *umount /nasdirektorij*
- Uporabniška imena: *std\miha*