Quick Reference Card 3 Course UNIX/Linux System & Network Administration

Your?
host logo

Partitioning

```
                              #Three example tools (as root):
fdisk     -l                  #List all disks and partitions
parted    /dev/hdb            #Partition slave disk on primary channel
cfdisk    /dev/sda            #Partition first serial disk
df        -h                  #Show disk free space (-human readable)
du        -h                  #Show disk usage (-human readable)
fdformat /dev/fd0H1440        #Low level floppy disk format
mkdosfs /dev/fd0              #High level floppy disk format
#LVM                          #Logical Volume Management (8e):
lvm                           #Show LVM commands
pvdisplay                     #Show physical volumes
vgdisplay                     #Show volume groups
lvdisplay                     #Show logical volumes
fdisk      /dev/hda           #Create partition type '8e', e.g. hda5
pvcreate /dev/hda5            #Create physical volume
vgcreate system /dev/hda5     #Create volume group 'system'
vgextend system /dev/hda6     #Extend volume group 'system'
lvcreate -L 4M -n data system #Create logical volume 'data' on 'system'
mkfs.reiserfs /dev/system/data #Format lv. Or e.g.: mke2fs -j /dev/sys...
vi /etc/fstab                 #Edit to survive a reboot
mount -a                      #Mount all stuff from /etc/fstab
lvextend -L +1G /dev/system/tmp#1) Resize logical volume. 2) Resize fs:
e2fsadm -L +1G /dev/system/tmp#     Resize xfs, ext3 and reiser:
xfs_growfs -d /tmp            #ext2online /dev/system/tmp
resize_reiserfs -s+1G /dev/system/tmp
#Create and use an LVM Snapshot Volume named 'snap':
mount   -o remount,ro /data
lvcreate          -s -L 5M -n snap /dev/system/data
mount   -o remount,rw /data
mount   /dev/system/snap /snapdir#Backup snapdir. Run after backup:
umount /snapdir
lvremove /dev/system/snap
#RAID                         #Software RAID (fd), raidtools package:
fdisk     /dev/sdd            #Create partition type 'fd', e.g. sdd1
fdisk     /dev/sde            #Create partition type 'fd', e.g. sde1
vi        /etc/raidtab        #Design /dev/md0; See: man raidtab
   persistent-superblock 1    #For raid-level 0, 1 & 5
mkraid  /dev/md0              #Create /dev/md0; Does a raidstart -a
(raidstop /dev/md0            #Aft changing raidtab and before mkraid)
mkfs.reiserfs /dev/md0        #Format disk set, or e.g.:
mke2fs  -j -b 4096 -R stride=8 /dev/md0
mount   /dev/md0  /data       #Don't forget /etc/fstab for the reboot
cat       /proc/mdstat        #Check raidset; lsraid -A -a /dev/md0
raidsetfaulty /dev/md0 /dev/sde1 #Test raidset
raidhotadd /dev/md0 /dev/sde1 #Test raidset
raidadd -a ...                #Regenerate raidset after break down
mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sda1 \
/dev/sda2                     #Alternative mdadm RAID frontend
#Swap partition               #Create swap partition (82)
mkswap /dev/hda6              #Format swap partition
swapon  /dev/hda6             #Enable swapping
vi        /etc/fstab          #Add swap mount
cat       /proc/swaps         #Check swap status
```

Unattended install

```
                              #Type at install boot prompt:
autoyast=nfs://ip/path/file.xml #Use nfs:, smb:, http: or ftp:
linux ks=floppy               #ks.cfg from /root/anaconda-ks.cfg:
   nfs --server 172.28.24.24 --dir /install/rhel3
```

Install/manage an application with rpm

```
rpm -qa                       #Display all applications (packages, rpm's)
rpm -ivh        file.rpm      #Install file.rpm; freshen,update only:-Fvh
rpm -Uvh        file.rpm      #Install or update an rpm
rpm -qf         file          #Show owner package of file
rpm -qi         rpm           #Show header info (or -qpi file.rpm)
rpm -qpl        file.rpm      #Show files owned by package
rpm -V          rpm           #Verify rpm
rpm -e          rpm           #Erase package
rpm --import    RPM-GPG-KEY#Key is on CD. Also run:
gpg --import    RPM-GPG-KEY#Now you can check integrity:
rpm --checksig file.rpm       #Verify rpm authenticity. Same as -K
rpm -Uvh        rpmdb-redhat  #Make auto solve dependencies possible:
rpm -ivh --aid file.rpm       #Requires RPMS CD's in current directory
for i in $( rpm -qa ¦ grep text ) ; do rpm -e $i ; done
```

Install from source

```
                              #Not supported with support
#Download the source (tarball), like file.tar.gz
tar -zxvf file.tar.gz         #Unpack the tarball in current directory
cd ......                     #Change to the new source directory
./configure                   #Create the Makefiles (rules for compiler)
make                          #Compile
make install                  #Put program into production environmnt
```

Install/manage an application with debian

debian

```
dpkg                          #Debian package file installer
dpkg-reconfigure              #Reconfigure installed package
apt-get                       #Advanced package tool (remote) installer
apt-get upgrade               #Automatic upgrade tool
aptitude                      #Optional text/command front end for apt
synaptik                      #Optional gui front end for apt
dselect                       #Menu driven apt/dpkg package manager
apt-rpm                       #Install the apt-way on rpm systems
alien or martian              #Convert rpm or tarball to/from deb
```

Automounting

```
                              #Automounting example for NIS-users:
#or use http://www.flyn.org/projects/pam_mount/
vi /etc/auto.master
  /home/guests  /etc/auto.guests  --timeout=60
vi /etc/auto.guests
  *   -rw,soft,intr 172.28.24.24:/home/guests/&
```

Quota Management

```
1) vi /etc/fstab              #Change mount options in fstab:
     /home ....  usrquota,grpquota
2) mount -o remount /home     #Activate changes by remounting
3) quotacheck -cM /home       #Create aquota.user & aquota.group, or:
   quotacheck -aguv           #all partitions, grp & usrquota, verbose
4) quotaon /home              #Prepare quota for restart at boot
   /etc/init.d/quota start    #Make sure quota management is started
   chkconfig -a quota         #Prepare for runlevel 2,3,5 after reboot
5) edquota -u username        #Edit usrquota (-u), -g is grpquota
#Check and test the quota status and limits (1 block = 1 kB):
quota
repquota -aguv
dd if=/dev/zero of=bigfile bs=1M count=5
du -a /home ¦ sort -nrb ¦ less #Find the biggest files
```

Quick Reference Card 3 Course UNIX/Linux System & Network Administration

Your?
host logo

Service file configuration examples

```
#Apache + virtual host + access control:
vi /etc/httpd/conf/httpd.conf or vi /etc/apache2/httpd.conf
  NameVirtualHost 172.28.24.24
    <VirtualHost 172.28.24.24>
      ServerName     wwwX.example.com
      ServerAdmin    root@stationX.example.com
      DocumentRoot   /var/www/wwwX/html
      Errorlog       logs/wwwX-errorlog
      Customlog      logs/wwwX-accesslog combined
      <Directory     /var/www/wwwX/html>
        Options      Indexes Includes
        Order        allow,deny
        allow from   .example.com
        AllowOverride AuthConfig
      </Directory>
    </VirtualHost>
vi /var/www/wwwX/html/.htaccess #Requires AllowOverride AuthConfig
  AuthName      "Whitehouse stuff"
  AuthType      Basic
  AuthUserFile  /etc/httpd/conf/wwwX.htpasswd
  require       valid-user
htpasswd -mc    /etc/httpd/conf/wwwX.htpasswd username
chgrp apache    /etc/httpd/conf/wwwX.htpasswd
chmod g+r       /etc/httpd/conf/wwwX.htpasswd
#FTP servers                          #User 'daemon' or 'ftp' plus homedir
vi /etc/vsftpd/vsftpd.conf            #/etc/init.d/vsftpd uses /var/ftp
  userlist_enable = yes               #Requires:
vi /etc/vsftpd.user_list              #Accept list
vi /etc/vsftpd.ftusers                #Deny list
vi /ftpdir/.message                   #Welcome file
vi /etc/pure-ftp/pure-ftpd.conf       #/etc/init.d/pure-ftpd
usermod -d /var/ftp  ftp              #Change ftp-directory
```

Secure Teleworking

```
                                      #Execute next commands at local host:
scp file  usr@remhost:/dir            #Secure copy to remote host (ip or name)
sftp C user@host                      #Secure ftp using ssh and compression
ssh usr@remhost                       #Secure telnet terminal
ssh-keygen -t dsa -b 2048             #Create rsa|dsa keys for automatic logon
#Put public key on the managed hosts:
scp  ~/.ssh/id_dsa.pub  usr@remhost:/tmp
ssh usr@remhost mkdir  ~/.ssh
ssh usr@remhost cat /tmp/id_dsa.pub >> ~/.ssh/authorized_keys
#To make life easier after every reboot:
ssh-agent bash                        #Add passphrase interception
ssh-add                               #Add passphrase to running ssh-agent
ssh -XC usr@remhost /sbin/yast2#Start remote X-client and display locally
#putty.exe and e.g. wiki.freedesktop.org/wiki/Xming for Windows
#Redirect Local port 23 via ssh to 10.0.0.2:23 using port 22:
ssh -L 23:10.0.0.2:23 -N usr@10.0.0.2 &
#MITM command to connect Remote myhome:23 to local network 10.0.0.1:23:
ssh -R 23:10.0.0.1:23 -N root@myhome &
#Terminal Server client               #See also ltsp.org. Enable remote access:
vi /etc/sysconfig/displaymanager      #Tip: Use YaST or other GUI-tool
  DISPLAYMANAGER_REMOTE_ACCESS="yes"  #on application server
X -query  172.28.24.24                #Gives local <alt-f7> for remhost (init 3)
X -query  172.28.24.25  :1.0  &       #Gives local <ctrl-alt-f8> for remhost
```

TCP wrapper

```
                                  #Check e.g. portmapper. Order &example:
vi /etc/hosts.allow               #See: man 5 hosts_access
  in.telnetd : ALL : spawn echo "login from %c to %s | mail -s warn root
vi /etc/hosts.deny
  sshd : ALL EXCEPT 172.28.24.
```

Super daemon xinetd

```
                                  #Use snippet name from /etc/services
vi /etc/xinetd.conf               #Edit general settings
vi /etc/xinetd.d/snippet          #Change settings per snippet
  only_from    = 172.28.0.0/16  172.27.200.1
  no_access    = 172.28.24.1      #or: access_times = 9:00-18:00
  redirect     = 10.0.0.2  80     #Redirect to newtargetip  newport
  log_type     = FILE /var/log/redirector.log
  wait         = no    | yes      #no=tcp, yes=udp
  protocol     = tcp   | dgram#tcp=tcp, dgram=udp
  socket_type  = stream           #tcp and udp
/etc/init.d/xinetd restart        #Activate changes
```

Logging

```
                             SuSE #Run SuSEconfig aft vi /etc/sysconfig/...
vi /etc/syslog.conf               #vi /etc/syslog-ng/syslog-ng.conf.in  SuSE
  #man syslog.conf facilit.&prior. #  options { group(loggroup); ...
vi /etc/sysconfig/syslog          #Edit syslogdaemon -r for remote logging
  SYSLOGD_PARAMS="-r -s my.domain" SuSE  🎩: SYSLOGD_OPTIONS
logger -i -t yourname "Text"      #Generate message for logfile
vi /etc/logrotate.conf            #Edit rotate, size. Also /etc/logrotate.d/...
```

Firewall

```
                                  #netfilter, 3 tables, 11 chains, e.g.:
vi firewall.sh                    #Create example firewall script
  #!/bin/bash
  iptables -F                     #Flush all rules
  iptables -P INPUT DROP          #Set default INPUT Policy
  iptables -A INPUT -i lo -j ACCEPT
  iptables -A INPUT -i eth0 -p tcp  --sport 22 -j ACCEPT
  /etc/init.d/iptables save     🎩#Save all rules
  iptables-save > file     🎩   SuSE#Save all rules; iptables-restore < file
  exit 0                          #Tip: portnr/tcp/udp is in /etc/services
iptables -nvL                     #List current rules (numeric, verbose)
service iptables on             🎩#Survive a reboot
#Ethereal                         #Free sniffer
tpc.port==80&&ip.addr==12.2.2.1   #Capture using example filters
telnet nu.nl 80                   #Check web availability with: get /
netcat -v -w 1 -z nu.nl 80        #Check web availability ('open')
```

Recovery from an unknown root password # 🎩 Use single user mode

```
#Use rescue mode (linux rescue) or CD-boot and remove x in /etc/passwd
```

Rebuild the kernel

```
                                  #Not supported with support
#Select what needs to be compiled from /usr/src/linux/:
make config | make menuconfig | make xconfig
make dep                          #Not for 2.6 kernel and higher
make clean                        #Get rid of old settings
make bzImage                      #/usr/src/linux/arch/i386/boot/bzImage
cp bzImage /boot/vmlinuz...       #Put kernel at the boot location
make modules                      #Building modules
make modules_install              #Installing modules
mkinitrd                          #Compile new /boot/initrd...
vi /boot/grub/menu.lst            #Edit bootmenu and add the new kernel
```