

<code>useradd &lt;username&gt;</code>	Create user
<code>passwd &lt;username&gt;</code>	
<code>useradd -m &lt;username&gt;</code>	(To create home directory for user)
<code>usermod -l &lt;newname&gt; &lt;oldname&gt;</code>	Rename username
<code>userdel &lt;username&gt;</code>	Delete user
<code>userdel -r &lt;username&gt;</code>	(To remove home directory)
<code>usermod -L &lt;username&gt;</code>	Lock user account
<code>whoami</code>	Show who is the current user
<code>id</code>	Display user information
<code>id &lt;username&gt;</code>	
<code>chage [-m mindays] [-M maxdays] [-d lastday] [-I inactive] [-E expiredate] [-W warndays] &lt;user&gt;</code>	Password aging policies
<code>chage -d 0 &lt;username&gt;</code>	Force password update on next login
<code>chage -l &lt;username&gt;</code>	Show current settings for user
<code>groupadd &lt;groupname&gt;</code>  <code>groupadd [-g &lt;uid&gt;] &lt;group&gt;</code> E.g. <code>groupadd -g 501 services</code>  Note: GID 0 → reserved for root group GID 1-499 → reserved for system and application use GID 500+ → allocated for user's group	Create group
<code>groupmod -n &lt;new groupname&gt; &lt;old groupname&gt;</code>	Rename group
<code>groupmod -o -n &lt;new groupname&gt; &lt;old groupname&gt;</code>	(if group is already in use)
<code>groupdel &lt;groupname&gt;</code>	Delete group
<code>cat /etc/passwd   grep &lt;filter&gt;</code> <code>cat /etc/shadow   grep &lt;filter&gt;</code>	Show all users

<code>usermod -a -G &lt;group&gt; &lt;user&gt;</code>	Add user to a group
<code>Gpasswd -a &lt;username&gt; &lt;group&gt;</code>	
<code>cat /etc/group   grep &lt;filter&gt;</code>	Show which user is in which group
<code>mkdir &lt;path&gt;</code>	Create directory
<code>rmdir &lt;path&gt;</code>	Remove directory
<code>ls -ld &lt;path of directory&gt;</code>	Listing/ show permissions of directory
<code>ls -l</code>	Listing/ show permissions
<code>ls -l &lt;path&gt;</code>	Listing/ show permissions of a file
<code>chgrp &lt;groupname&gt; &lt;path of directory&gt;</code>	Add group to directory
<code>chown &lt;user&gt; &lt;file&gt;</code>	Change file owner
<code>chown &lt;user&gt;:&lt;group&gt; &lt;file&gt;</code>	Change owner and group of a file
<code>chgrp &lt;group&gt; &lt;file&gt;</code>	Change file owner to a group
<code>chmod &lt;user permission&gt;&lt;group permission&gt;&lt;other permission&gt; &lt;filename&gt;</code>  E.g. chmod 770 <filename> gives owner rwx, group rwx and the world ---	Change permissions of a file (absolute/numeric mode)
<code>chmod &lt;u/g/o/a&gt;=&lt;permission&gt; &lt;filename&gt;</code> E.g. chmod ug=rwx <filename>  <code>chmod &lt;u/g/o/a&gt;&lt;+/-&gt;&lt;permission&gt; &lt;filename&gt;</code> E.g. chmod ug+rwx <filename>  u = users, g = group, o = others, a= all	Change permissions of a file (symbolic mode)
<code>chmod u+s &lt;filename&gt;</code>	Set setuid
<code>chmod u-s &lt;filename&gt;</code>	Remove setuid
<code>chmod g+s &lt;directory name&gt;</code>	Set setgid
<code>chmod g-s &lt;directory name&gt;</code>	Remove setgid
<code>chmod &lt;setgid/setuid&gt;&lt;user permission&gt;&lt;group permission&gt;&lt;other</code>	Set setuid and setgid permissions

permission> <filename>  E.g. chmod 6711 <filename/directory name> 4: setuid, 2: setgid, 6:both	
chmod +t <directory name>  chmod 1<r/w/x> <directory name> e.g. chmod 1777 london	Set stickybit
cd <path> echo <text> > <filename>  cd <path> echo <text> >> <filename>	Create file (overrides text in file)  (adds text into file)
touch <filename>	Create file with no text inside
mount -o remount, rw <filename>	Mount file system
quotacheck -cugfm /<directory>	Run quotacheck program
quotaon /<directory>	Start quota
quotaoff /<directory>	Stop quota
edquota <username>	Edit quota
Getfacl <filename/directory name>	View ACL of file or directory
setfacl -m u:<user>:<permission> /<filepath>	Add ACL permissions for user to file
setfacl -m g:<group>:<permission> /<directory path>	Add ACL permissions to user to a group
Setfacl -dm "entry" /<directory path>	To allow all files or directories to inherit ACL entries from the directory it is within
setfacl -x "entry" /<directory or file path>	Remove specific entry
setfacl -b /<directory or file path>	Remove all entries
setfacl -m d:o:rx <directory path>  Must put d: for default directory ACL	Default directory ACL
setfacl -m f:o:rx <file path> Must put f: for default file ACL	Default file ACL

ls -l <directory path> E.g. -rw-rwx-r--+	Check ACL status (look for the + sign at the end)
tar [options] <destination> <source>	tar syntax
tar -cpvzf <destination> <source>  E.g. tar -cpvzf /tmp/backup.tar /home  c = create a new archive v = verbose p = preserve the permissions z = compress the tar archive using gzip (aka zip the files) f = specify name of archive file	Backup directory using tar
tar -tzf <filename>  E.g. tar -tzf /tmp[/backup.tar  t = list the contents of the archive z = uncompress the tar archive using gzip f = specify name of archive file	List archived files using tar
tar -xvzf <file you want> -C <copy location>  E.g. tar -xvzf /tmp/backup.tar -C /tmp/recover  x = extract files from the archive z = uncompress the tar archive using gzip f = specify name of archive file v = verbose <b>Note: /tmp /recover must exist</b>	Recover files using tar
rsync [options] <source>/ <destination>	rsync syntax
rsync -avz <source>/ <destination>  E.g. rsync -avz /home/student/ /tmp/studentbackup  a = archive mode; equals rlptgoD (no H, A, X) v = verbose z = compress mode during transmission <b>Note: / at the end of the source. This is to tell rsync that the source is</b>	Backup using rsync (incremental backup)

<p>all the contents of the source directory. If the "/" at the end of the source directory is missing, rsync will simply create a copy of the source directory instead of its contents.</p> <p>If / tmp studentbackup does not exist, it will be created</p> <p>You must have the necessary permissions to copy</p>	
su -l <name>	Log in as another user
cd /	Change to root directory
cd ~	Change to home directory
cd ..	Change to 1 level up (directory)
cd -	Go back to previous directory
man	Man page

Cat shows whole file. Grep finds keywords and shows that line

ACL info: <https://www.shrubbery.net/solaris9ab/SUNWaadm/SYSADV6/p50.html>

grep is to find a word/regex in a file or whatever that is pipe to it

cat is to print the whole file

Press q to quit man page

Stored in /etc/login.defs	Default password policy
---------------------------	-------------------------

Setuid:

- setuid are needed for tasks that require higher privileges than those which common users have.
- When setuid bit is set, an executable when launched does not run with the privileges of the user who launched it, but with that of the file owner instead.
- For example, if an executable has the setuid bit set on it, and it's owned by root, when launched by a normal user, it will run with root privileges.

#### Setgid:

- Used to create a collaborative directory
- When a file is created in a directory with the setgid bit set, it belongs to the same group as the group directory, rather than the creator's primary group
- setgid affects both files as well as directories.
- When used on a file, it executes with the privileges of the group of the user who owns it instead of executing with those of the group of the user who executed it.

#### Setgid on directory:

- Causes new files and subdirectories created within it to inherit the groupID of the folder, rather than the primary groupID of the user who created the file.
- Newly created subdirectories inherit the setgid bit. Existing entities (files/subdirectories) will not be affected.
- This is used for file sharing since they can be now modified by all the users who are part of the group of the parent directory.

#### Sticky bit

- If not set, users with write permissions to a directory can delete any file in that directory regardless of that file's permissions or ownership.
- When set, only the item's owner, the directory's owner, or the root can rename or delete a file under the directory.
- Typically it is set on the /tmp to prevent ordinary users from deleting or moving other users' files.
- Members cannot delete other members' files.
- In Unix notation, the sticky bit is represented by the letter t in the final character place.
- If the sticky bit is set on a directory without the execution bit set for the others category, it is indicated with a capital T.

#### Disk Quota (Week 12, Practical 10B):

- 1) Modify /etc/fstab (must be mounted with usrquota or grpquota options)
- 2) Unmount and mount file system
- 3) Run quotacheck program
- 4) Run endquota to set up user quota

Permissions:

Number	Permission Type	Symbol
0	No Permission	---
1	Execute	--x
2	Write	-w-
3	Execute + Write	-wx
4	Read	r--
5	Read + Execute	r-x
6	Read +Write	rw-
7	Read + Write +Execute	rwx

UID Range:

0 → root, has special privileges

1-499 → systems users, non-interactive service accounts

500+ → regular users with interactive access to machine

Command summary:

groupadd - create a new group.

groupdel - delete a group.

groupmod - modify a group

useradd - create a new user account or update default new user information

userdel - delete a user and related files

usermod - modify a user account

chgrp - changes group ownership of files

chown - change owner of file(s) to a different user

passwd - sets user's password

chage - used to change time that the user's password will expire/ password aging policies

mkdir - create directory

rmdir - remove directory

chmod - modify permissions of file/directory

Reference to practical:

- 1) Create user, group, add members (Week 11, Practical 9C (Terminal), Practical 9D (GUI))
- 2) Linux basic commands - make directory, create files, list directory (Week 11, Practical 9A (this is a basic command list))
- 3) Add hard disk etc.. (Week 8, Practical 8C)
- 4) Shared folder (Week 12, Practical 10A)
- 5) Set file permission and ACL (Week 12, Practical 10A; Week 13, Practical 11)
- 6) Network configuration (Week 8, Practical 8B)
- 7) Backup (Week 13, Practical 11)

Overall view of linux commands (Week 11, Practical 9A)

Group info (Week 11, Linux Groups)

Password info (Week 11, Linux Passwords)

Disk quota (Week 12, Practical 10B)

Practical 11 Exercise Answers (Week 13)

All answers have to be found inside practicals 8-11 for linux

Reference to practical:

- 1) Create OU, Group, User, password policy. (Week 2, Practical 2A, Practical 2B; Week 6, Practical 6A)
- 2) Raid 0 to 5 (Week 3, Practical 3)
- 3) Add hard disk (Go to VM settings to add)
- 4) Roaming profile (Week 1, Practical 1D)
- 5) Shared folder - NTFS and shared permissions (Week 4, Practical 4A)
- 6) Disk quota (Week 3, Practical 3)
- 7) Set encryption, mount drive (Week 4, Practical 4B; Week 3, Practical 3)
- 8) Group Policy (Week 2, Practical 2B)
- 9) Remote access (Week 5, Practical 5A)
- 10) Network configuration (Right-click the network icon in the notification area (bottom right of screen), and then click Open Network and Sharing Center → change adapter settings → right click ethernet0 → click on properties → click on Internet Protocol Version 4 (TCP/IPv4) → Properties → Manually enter the configuration)

Promote windows server to domain controller (Week 1, Practical 1B)

RAID 0: Striped volume (need 2 disks)

RAID 1: mirrored volume (need 2 disks)

RAID 5: need 4 disks

3 way mirror: need 5 disks

RAID 6: striping with double parity (need 7 disks)  
RAID 1+0: striping with mirror (need 4 disks)  
Change file attributes/compress/file encryption (Week 4, Practical 4B)  
Backup and restore (Week 5, Practical 5B)  
Security Template (Week 6, Practical 6A)  
Applocker or Bitlocker (Week 6, Practical 6B)  
Firewall (Week 7, Practical 7)

<https://github.com/zaramichaela/SystemsAdministrationAndSecurity>