OPERATING SYSTEMS
CCGC-5000

Module - 04

Agenda

HUMBER
Faculty of Applied Sciences & Technology

Authentic information is available from the given resources in course outline and URL's mentioned from this slides, and this presentation is only supportive document to be read with the given resources and corrected accordingly if required..

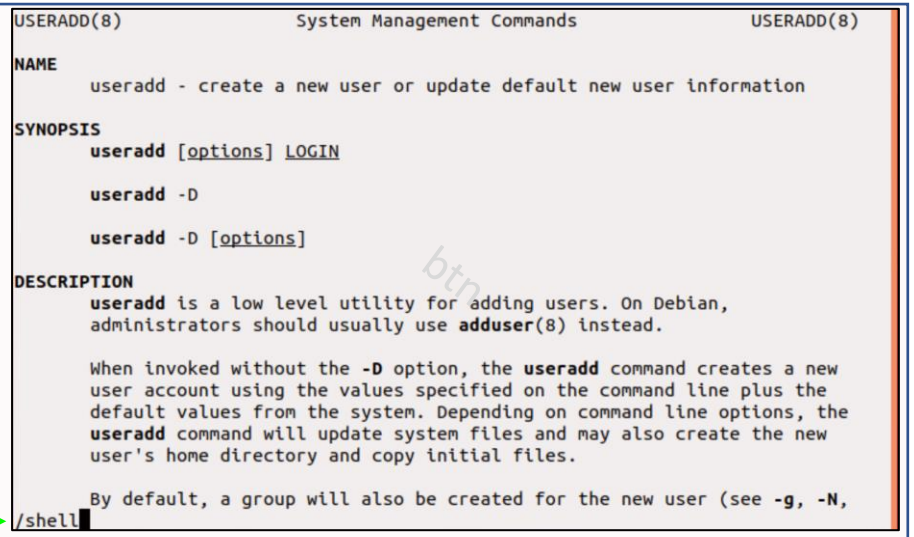
- Bash Scripting using read
- User management
- User Accounts
  - Super user
  - Regular user
  - System user
- User management files
- User Management Tools
- Password ageing
- Groups
  - Primary
  - Supplementary
- Group management
- Sudoers

Must read

- Chapters 5,6 of RHEL8, 2nd Edition book
- RedHat documentation https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/

Controlling viewing of Manual pages

- When viewing manual pages of a command, the display can be controlled as below:
- Manual pages moves a full screen if spacebar is pressed
- Also **PageUp** and **PageDown** key will move one full screen up and down respectively
- **Up arrow key** and **Down arrow key** can be used to move one line up and down respectively
- To exit the display, just type **q**.
- To search a word in manual pages, type **/** followed by the word to search and press enter
- To go to next word of the search, just type **n** like wise pressing n will go the occurance of next word until the last one.
- If the word in not available of after last word is found it will display Pattern not found



*above in **man useradd** command, to search for word shell*

read command

HUMBER
Faculty of Applied Sciences & Technology

- **read** is a shell built in command which helps to read the standard input and assign the input to the given variable
- **read** command is used in bash script to get input, store in variable and display it
- Following are 3 examples with read command using bash script.
  1. **using echo and read**
  2. **using echo -n and read**
  3. **using read -p instead of echo -n and read**
- Note in all the three examples output are **SAME**

```
[user1@hostname ~]$ read var1
Hello World
[user1@hostname ~]$ echo $var1
Hello World
```

```
[user1@hostname ~]$ read var2
345
[user1@hostname ~]$ echo $var2
345
```

bash script (example - 1)

```
#!/bin/bash
echo "Enter part number :"
read pnum
echo "Part number entered is $pnum"
```

executing bash script

```
Enter part number :
12345
Part number entered is 12345
```

bash script *using -n option with echo* (example -2)

```
#!/bin/bash
echo -n "Enter part number :"
read pnum
echo "Part number entered is $pnum"
```

executing bash script

```
Enter part number :12345
Part number entered is 12345
```

bash script *using **read -p*** (example - 3)

```
#!/bin/bash
read -p "Enter part number :" pnum
echo "Part number entered is $pnum"
```

executing bash script

```
Enter part number :12345
Part number entered is 12345
```

UNIX Linux User Management

HUMBER
Faculty of Applied Sciences & Technology

- **User Management and Administration**

- Create, modify, suspend, deactivate, activate, delete users
- Maintain/reset users/passwords
- Allocating and managing /home directories
- Making files available in home directory for new users
- Compliance of password policies
- Applying effective security policies
  - File and directory permissions
  - Disk quota
- Create/modify/delete and manage groups for users of same permissions and activities
- Add/remove users from groups

- **Linux/UNIX USERS (USER ACCOUNTS)**

- System identifies users and groups by numbers known as user ID (UID) and group ID (GID)
- Every user and group is assigned with UID *(userID)* and GID*(groupID)* respectively, and there are 3 types of linux/unix users :

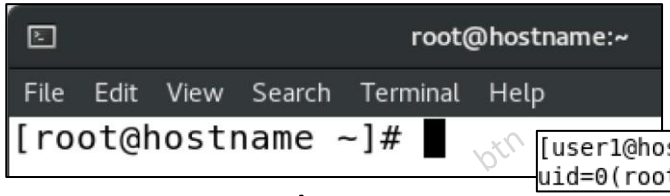
- **Super User #**
  Has total and complete control over all aspects of the system with UID and GID being zero commonly referred to as **root**

- **Regular user**
  Use the system for Non-administrative tasks (example : johns, jsmith, etc.,)

- **System user**
  An administrative account that the system uses for running varous day-to-day services (example : www-data, lp, mail, etc.,)

Super User

HUMBER
Faculty of Applied Sciences & Technology



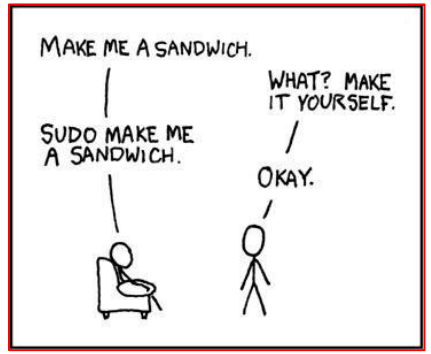

```
[user1@hostname ~]$ id root
uid=0(root) gid=0(root) groups=0(root)
```

- Unique user account having supreme power
- Most commonly referred as **root** user
- Unrestricted access to file system, permitted to do any operation on the system
- Grant and revoke access to files & directories
- Locked by default in Ubuntu
- Unlock the root user by assigning password to root user
- **UID – zero** for root user

```
[user1@hostname ~]$ id root
uid=0(root) gid=0(root) groups=0(root)
```

- When logged in as root, the command prompt ends with #

```
root@lakeshore:~#
```



- Using ***sudo*** with a command, the command is executed as root user. *(example: sudo adduser username, sudo passwd username, etc.,)*

- **su, su -, su root** command change user and login as root or become root user after entering root user password *(**su username** can be used to change from current user and login to another user)*
- **sudo -i** command logs in as root after entering the currently logged in user password provided the user has administrative privilege.
- In RHEL, for a user to use **sudo** with a command, the user has to be member of **wheel** group preferably supplementary group
- In Ubuntu, for a user to use **sudo** with a command, the user has to be member of **sudo** group preferably supplementary group

Super User ...*contd.,*

- Not recommended to be logged in as root user
- When logged in as **root** user, the user has to be very cautious, due to extreme super power of root user.
- Loss or damage caused as root user can be catastrophic
- Always login as a regular user and only use super user temporarily to do specific administrative tasks
- For more security, remote login of root need to be disabled.
- Ubuntu disables root login by default
- Use **sudo,** only when permission is denied
- There is **NO** necessity to use **sudo when managing files in your own home directory.**

Handle with care !

Regular User

- User who logs in to do his daily tasks
- In RHEL, the **UID** of regular user **starts from 1000**
- Regular user, normally has its home directory with its name in /home directory. *(example /home/username)*
- These users do not need to make system-wide changes or manage other users.
- Mostly perform non-administrative tasks unless provided with specific admin tasks
- Able to change settings specific to their own accounts
- Depends on the level of rights or permissions provided

System User

- Not related to any person but rather an administrative account used by system to run various day-to-day services
- System user **UID** is from **1 to 499** and **65534**
- System users **do not have**
  - home directory
  - Password
  - Permit access to the system through login prompt
- Ex: **www-data** system user owns Apache web server and all associated files
  - Only this user and root have access to these files

User management files

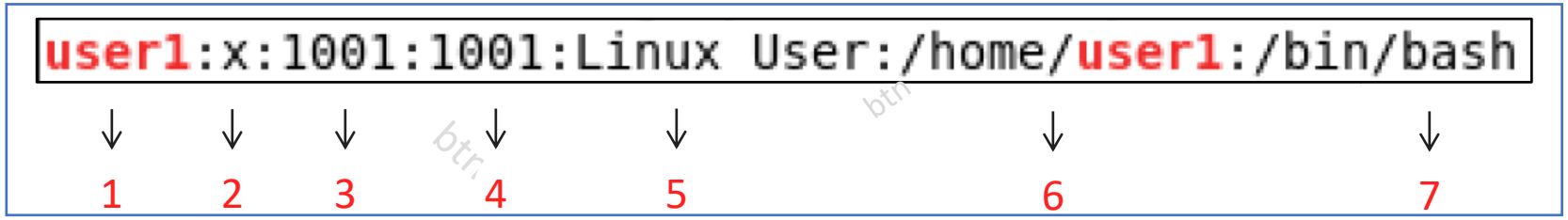
HUMBER
Faculty of Applied Sciences & Technology

- **/etc/skel** (directory)
  - When creating user, all the files in /etc/skel directory is copied to the new user's home directory.
- UNIX/Linux stores user and group information in the following files as a database
- /etc/passwd
  - It is database file to store information on all user accounts that are present on the system
  - You can find a list of all the users on a system
  - Recommended reading **man 5 passwd**
- /etc/group
  - It is also a database file which stores all the groups in the system
  - Recommended reading **man group**
- **/etc/shadow**
  - all users encrypted password and password aging information is stored in this file as a database
  - It is read only by root and Pluggable Authentication Modules(PAM) authentication manager
  - This file permission should be such that it is readable and writeable by root
  - Recommended reading **man shadow**
- **/etc/gshadow**
  - contains shadowed information for group accounts
  - This file must not be readable by regular users if password security is to be maintained
  - Recommended reading **man gshadow**

```
-rw-r--r--. 1 root root 1155 Feb  2 22:19 /etc/group
----------. 1 root root  928 Feb  2 22:19 /etc/gshadow
-rw-r--r--. 1 root root 2972 Feb  2 22:18 /etc/passwd
----------. 1 root root 2061 Feb  2 21:28 /etc/shadow
```

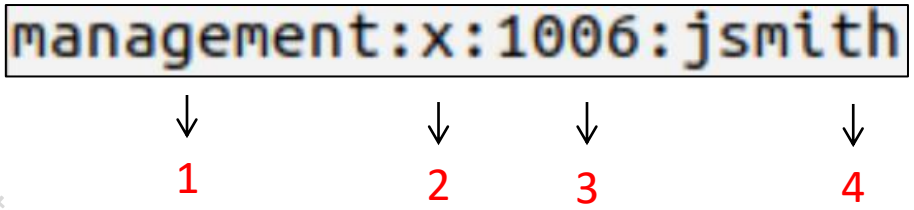
/etc/passwd

- **/etc/passwd** file one line for each user account, with seven fields delimited by **:** (*colon*) and the comment field is further delimited by , (comma)
- The fields are
  1. login username,
  2. password field
     - *optional encrypted password*
     - If **x** then password is stored in /etc/shadow file instead
  3. UID
  4. GID
  5. Username or Comment
  6. User Home Directory
  7. User login shell

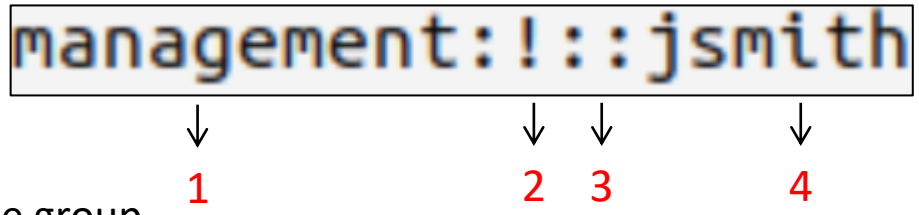


/etc/group & /etc/gshadow

- **/etc/group** file there is one entry per line for each group delimited by colon :
- The fields are
  1. groupname field
  2. password field
     - if x password stored in /etc/gshadow
     - if ! or *, users will not be able to use unix password to access the group
     - group members do not need passwd
  3. GID
  4. users added to the group separated by comma



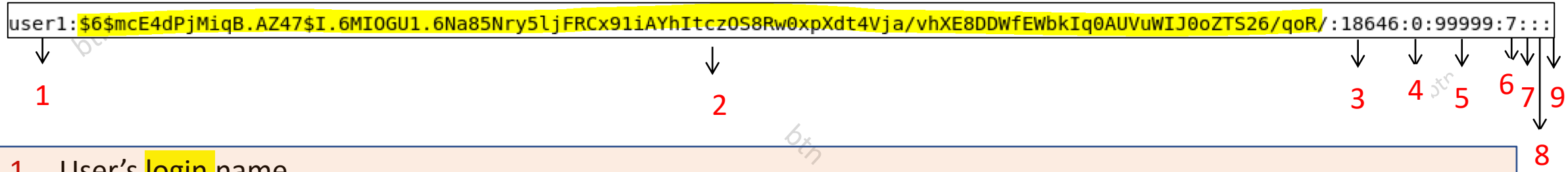
- **/etc/gshadow** file there is one entry per line for each group delimited by colon: and the fields are
  1. group name field
  2. encrypted password
     - if ! or *, users will not be able to use unix password to access the group
  3. administrators - comma separated list of user names
  4. members - comma separated list of user names



/etc/shadow

HUMBER
Faculty of Applied Sciences & Technology

- System users encrypted password and optional aging information is stored in **/etc/shadow**
- Each user's password info is stored in each line with 9 fields separated by colon the delimiter



1. User's login name
2. encrypted password for the user
   - password filed starting with ! means the password is locked
3. date of last password change expressed as number of days since 1 Jan 1970
   - if value is 0, then user need to change password at next login
4. minimum password age - days before password can be changed
5. maximum password age - days after which password must be changed
6. password warning period - days before expiry of password to warn user
7. password inactivity period - days after the password expires that the account is disabled
8. account expiration date - date of expiration of account expressed as the number of days since January 1, 1970
9. Reserved and not currently allotted for any use

User Management Tools/Commands

HUMBER
Faculty of Applied Sciences & Technology

- Wherever a user is created, the files updated are **/etc/passwd**, **/etc/groups** and **/etc/shadow**
- Command **useradd** *username* can be used to create user, but the following options could be required *(refer useradd --help for all the options)*
  - -**m** to create user home directory
  - -**s** to specify the default login shell for user
  - -**c** to specify the user's full name
  - -**g** to specify primary group
  - -**G** to specify the supplementary group

When a new user is created all files in **/etc/skel** directory will be copied to new users home directory

To find the defaults available for user creation can be found using **useradd -D** and default value stored in /etc/default/useradd

```
[user1@hostname ~]$ useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
```

- To create a user with *(this is an example)*
  - username **jsmith**
  - fullname as **John Smith**,
  - **primary group** finance and
  - **supplementary group** management
  - *by default shell is /bin/bash and home directory is in /home/username*
- Command to create user jsmith:

```
sudo useradd -c "John Smith" -g finance -G management jsmith
```

- To check if user is created:

```
[user1@hostname ~]$ grep jsmith /etc/passwd
jsmith:x:1006:1008:John Smith:/home/jsmith:/bin/bash
```


Must read manual pages of useradd command**, files:** /etc/passwd, /etc/group, /etc/shadow, /etc/gshadow

User Management Tools/Commands

HUMBER
Faculty of Applied Sciences & Technology

- An user account can be modified using **usermod** command
- usermod options: *refer usermod --help for more options*
  - **-c** comment (GECOS field)
  - **-g** primary group
  - **-G** supplemental group
  - **-a** append supplemental groups
  - **-L** lock the user account
  - **-U** unlock the user account
  - **-s** shell

```
[user1@hostname ~]$ sudo useradd jdoe
[user1@hostname ~]$ grep jdoe /etc/passwd
jdoe:x:1007:1010::/home/jdoe:/bin/bash
```

*If **no primary group** given during user creation, system creates a group with same name as user and defaults it as primary group for the user.*

- change primary group of user jdoe from group **jdoe** to **sales**
  (GID 1010 changed to 1011)

```
[user1@hostname ~]$ grep jdoe /etc/passwd
jdoe:x:1007:1010::/home/jdoe:/bin/bash
```

```
[user1@hostname ~]$ sudo usermod -g sales jdoe
[user1@hostname ~]$ grep jdoe /etc/passwd
jdoe:x:1007:1011::/home/jdoe:/bin/bash
[user1@hostname ~]$ grep sales /etc/group
sales:x:1011:
```

- Add comment (GECOS field)

```
[user1@hostname ~]$ grep jdoe /etc/passwd
jdoe:x:1007:1011::/home/jdoe:/bin/bash
```

```
[user1@hostname ~]$ sudo usermod -c "JohnDoe" jdoe
[user1@hostname ~]$ grep jdoe /etc/passwd
jdoe:x:1007:1011:JohnDoe:/home/jdoe:/bin/bash
```

- To change user information field **chfn** command could be used with various options available with it (*Refer **chfn --help** for options*)

```
[user1@hostname ~]$ sudo chfn -f "John Doe" jdoe
[user1@hostname ~]$ grep jdoe /etc/passwd
jdoe:x:1007:1011:John Doe:/home/jdoe:/bin/bash
```

Password

- Command **passwd** is used to assign or reset a password to an user account
- To **assign** or **reset** a password to an user account the command is
  passwd *username*
- Whenever passwd command is used for an user account **/etc/shadow** file is updated
- An effective password policy is a fundamental part of good system administration plan
- The policy should cover
  - Allowed and forbidden passwords
  - Frequency of mandatory password changes
  - Retrieval or replacement of lost or forgotten passwords
  - Password handling by users

```
[user1@hostname ~]$ sudo passwd jdoe
Changing password for user jdoe.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

passwd command & *password ageing*

HUMBER
Faculty of Applied Sciences & Technology

- Apart from assigning or resetting a password of an user account, the passwd command can also be used to

- lock an user **passwd –l** *username*
- unlock an user **passwd –u** *username*

- display status of user password **passwd –S** *username*

```
[user1@hostname ~]$ sudo passwd -S jdoe
jdoe PS 2021-02-08 0 99999 7 -1 (Password set, SHA512 crypt.)
```

username | Password status | Last Password change date | Min days | Max days | Warn days | Inactivity period

- delete user password **passwd –d** *username*
- expire user password **passwd –e** *username*
  forces user to change password at next login

- password expire warning days **passwd –w #** *username*
- password change min days **passwd –n #** *username*
- password change max days **passwd –x #** *username*

*has to be replaced with the number of days*

- The status information consists of 7 fields.
- The first field is the user's login name.
- The second field indicates if the user account has a locked password (**LK**), has no password (**NP**), or has a usable password (**PS**).
- The third field gives the date of the last password change.
- The next four fields are the minimum age, maximum age, warning period, and inactivity period for the password. These ages are expressed in days.

Password Age Management

HUMBER
Faculty of Applied Sciences & Technology

- Command **chage** helps to change password ageing for the user account *(change user password expiry information)*
- Command **chage -l** *username* displays user password expiry information.

```
[user1@hostname ~]$ sudo chage -l jdoe
Last password change                                    : Feb 09, 2021
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
```

- To change Min days to 1 : `[user1@hostname ~]$ sudo chage -m 1 jdoe`
- To change Max days to 30 : `[user1@hostname ~]$ sudo chage -M 30 jdoe`
- To change Warn days to 3 : `[user1@hostname ~]$ sudo chage -W 3 jdoe`

```
[user1@hostname ~]$ sudo chage -l jdoe
Last password change                                    : Feb 09, 2021
Password expires                                        : Mar 11, 2021
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 1
Maximum number of days between password change          : 30
Number of days of warning before password expires       : 3
```

- More available options can be learned using chage --help

User Management Tools / Commands

- To delete user, **userdel** can be used

  userdel *username*

- To delete user, its home directory and mail spool file

  userdel –r *username*

- User once created in system may not be deleted, rather the user is assigned an expiry date and account could be locked.

Groups

- Groups can be created and include Users in the groups for providing same permissions or to do similar activities in the system
- Each group created is assigned an number called Group ID (GID)
- Groups makes managing users a lot easier
- Setting group permissions enables
  - To setup work-spaces for collaborative working
  - Limiting access to system resources to only those users who need them
- A group cannot be a member of another group in Ubuntu
- Ubuntu uses a scheme UPG (*user private group*) in which the default is that group name is same as user
- Administrators or users with user creation rights can create users/groups and add users to the relevant groups
- User's group can be classified as **primary** and **supplementary** (or secondary) groups
- Apart from primary, other groups can be assigned to the user to have permission based on the groups and classified as **supplementary** groups

Groups: Primary

- Each User account has unique primary group
- If no primary group specified during user creation, system creates a group same as username and adds the user to the group
- By default the file's group will be user's primary group
- When creating user **-g** option is used to specify primary group with useradd command

  useradd -g *groupname username*
- An existing user can be modified and assign primary group **usermod -g** *groupname username*
- To find the primary group, /etc/passwd and /etc/group files are required
- Find the GID of the user in /etc/passwd and find the group name in /etc/group using the GID.

```
[user1@hostname ~]$ grep jsmith /etc/passwd
jsmith:x:1006:1008:John Smith:/home/jsmith:/bin/bash
[user1@hostname ~]$ grep 1008 /etc/group
finance:x:1008:
```

Supplementary

- Apart from primary, other groups can be assigned to the user to have permission based on the groups and classified as **supplementary** *(also called as **secondary**)* groups
- User can be member of more than one supplementary group
- When creating user **-G** option is used to specify supplementary group with useradd command

  useradd -G *groupname username*
- To add first supplementary group for an user account,

  usermod -G *groupname username*
- An user account if need to be member of more than one supplementary group, need to use **-a** option with **usermod** command

  usermod - -a G *group name username*
- /etc/group displays supplementary group of the user

```
[user1@hostname ~]$ grep jsmith /etc/group
management:x:1009:jsmith
```

Group management

- To create new group **groupadd** can be used

  groupadd *groupname*

```
[user1@hostname ~]$ sudo groupadd hrd
[user1@hostname ~]$ grep hrd /etc/group
hrd:x:1012:
```

- Remove an existing group

  groupdel *groupname*

- To provide group password

  gpasswd *groupname*

- To modify user's primary group

  usermod -g *groupname username*

- To add user's first supplementary group

  usermod -G *groupname username*

- To add user's subsequent supplementary group

  usermod -aG *groupname username*

- To remove user from the user's supplementary group

  gpasswd -d *username groupname*

User Management Tools/commands

- Print real and effective user and group IDs : **id**
- Command **id** displays the currently logged in users UID, GID, groups and securiety context
- To print a user's UID and GID: **id** *username*

```
[user1@hostname ~]$ id jsmith
uid=1006(jsmith) gid=1008(finance) groups=1008(finance),1009(management)
```

- Lists logged in users

  users

- Change password in batch

  chpasswd

- Update and create new users in batch

  newusers

- GUI interface can be used to manage users by selecting Settings -> Details ->Users

Refer the respective commands manual pages for available options and additional information about the commands.

sudoers

- The /etc/sudoers file allows particular users as defined in the file to run various commands as the root user, without needing the root password
- The file is composed of aliases (basically variables) and user specifications (which control who can run what)
- **sudo visudo** to edit /etc/sudoers file, to add or change permissions but has to use it with extreme caution
- **/etc/sudo.conf** file is configuration for sudo front end, which specifies the security policy and I/O logging plugins, debug flags as well as plugin-agnostic path names and settings.
- sudo consults the sudo.conf file to determine which policy and I/O logging plugins to load.
- If no **sudo.conf** file is present, or if it contains no Plugin lines, sudoers will be used for policy decisions and I/O logging.

Refer to man sudoers, man sudo.conf, man visudo