

# Users and accounts

Chonnam National University  
School of Electronics and Computer  
Engineering

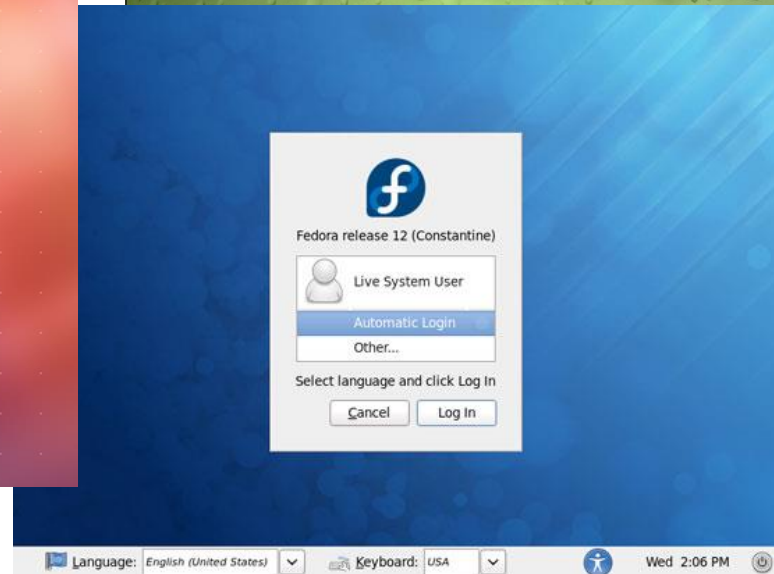
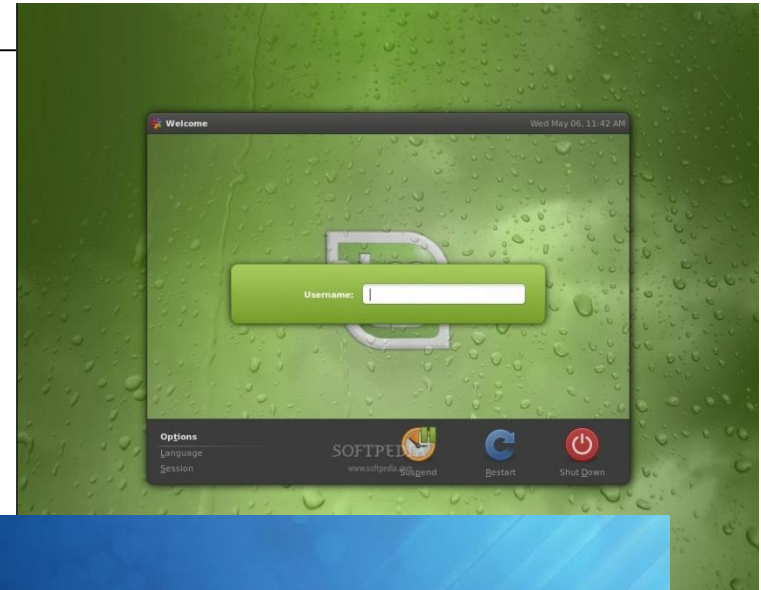
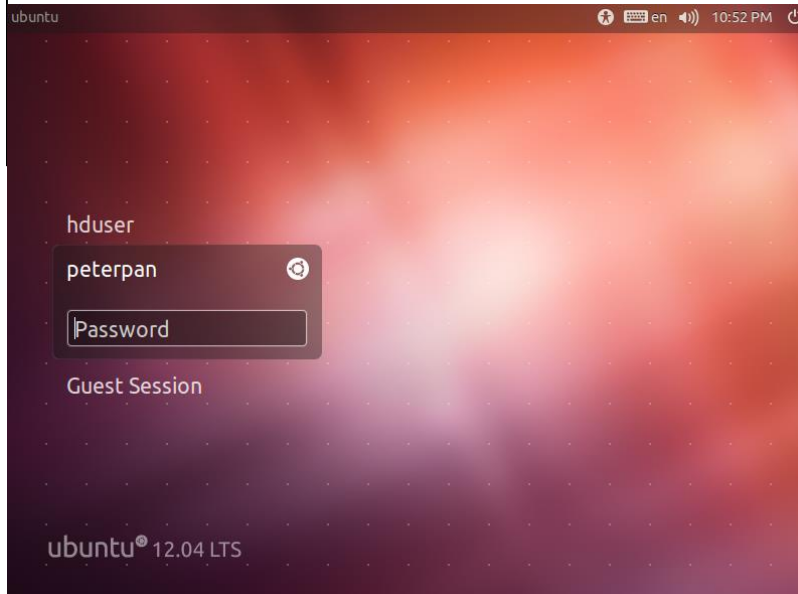
Kyungbaek Kim

# Using a LINUX system

- Login prompt displayed
  - When Linux first loads after booting the computer
  - After another user has logged out
- Need to enter a **username** and **password**
- The login prompt may be graphical or simple text
  - if text, login prompt will present a **shell**
  - If graphical, login prompt will present a desktop
    - A shell runs in a terminal window

# Login Prompts

```
Fedora release 13 (Goddard)  
Kernel 2.6.33.3-85.fc13.i686.PAE on an i686 (tty2)  
localhost login: _
```



# Linux Command Line

- A shell is where commands are invoked
- A command is typed at a shell prompt
  - A prompt ends in a sign : **\$** or **%** or **>**
- After typing a command, press **Enter** to invoke it
  - The shell will try to obey the command
  - Another prompt will appear
- Example:

```
$ date
Fri Mar 2 09:10:00 PST 2012
$
```

# Command Syntax

- Most commands take **parameters**
  - Some commands require them
  - Parameters are also known as **arguments**
  - Commands are **case-sensitive**
  - Example : echo simply displays its arguments

```
$ echo
```

```
$ echo Hello linux
```

```
Hello linux
```

```
$ ECHO
```

```
bash: ECHO : command not found
```

# Logging out

- To exit from a shell, use the **exit** command
- Pressing **Ctrl+D** at a shell prompt will also quit the shell
- Quitting all programs should log you out
  - In a text-only single-shell environment, exiting the shell should be sufficient
  - In a window environment, the window manager should have a log out command for this purpose
- After logging out, a new login prompt should be displayed
- C.F.)Shutdown : power off the machine

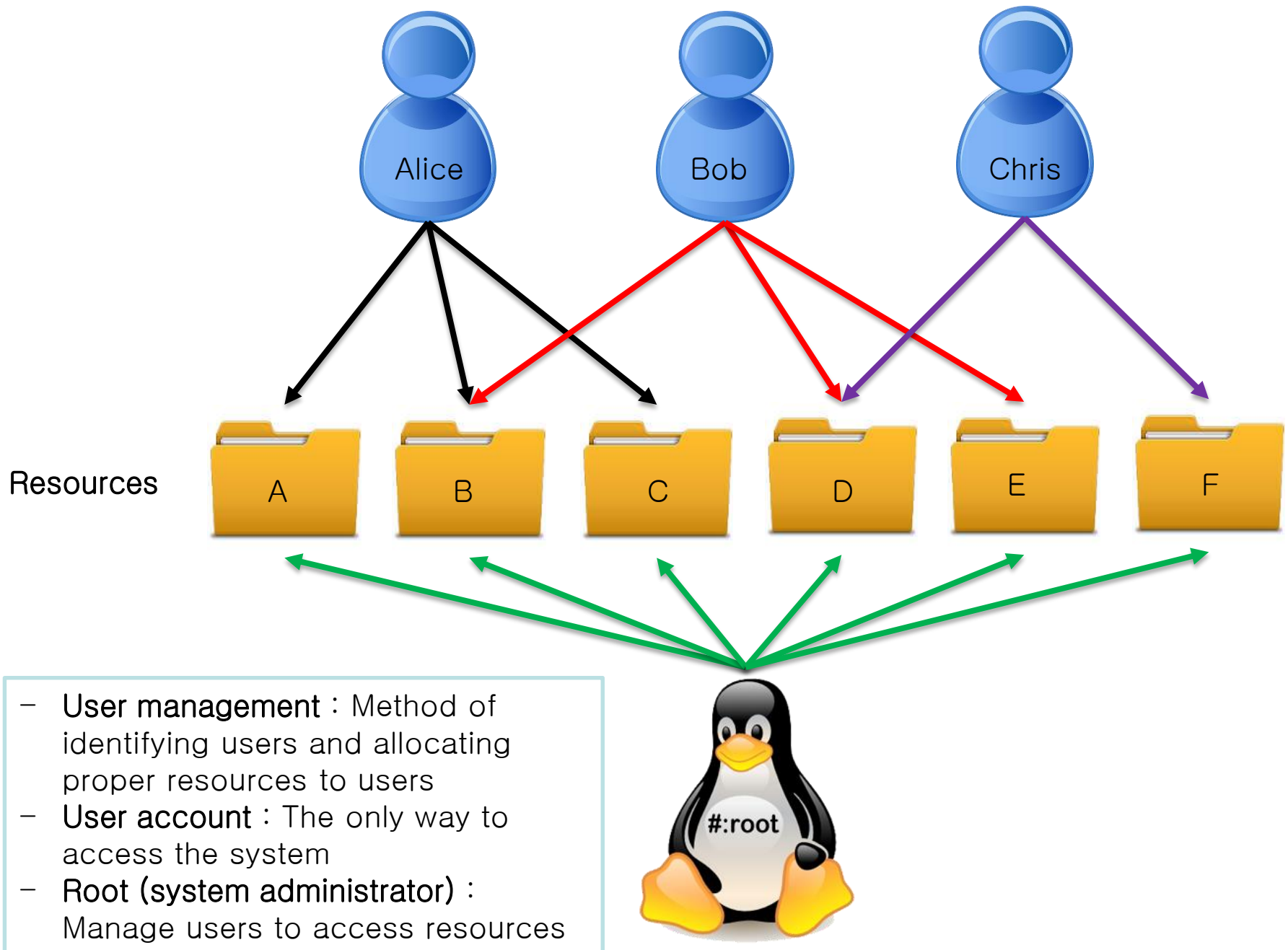
# Users and Groups

- Anyone using a Linux computer is a **user**
- The system keeps track of different users, by **username**
  - Security features allow different users to have different privileges
- Users can belong to **groups**
  - Allowing security to be managed for collections of people with different requirements

# The superuser : Root

- Every Linux system has a user called 'root'
- The root user is all-powerful
  - Can access any files
- The root user account should only be used for system administration, such as installing software
- When logged in as root, the shell prompt usually ends in '#'





# User accounts

account:password:UID:GID:GECOS:home directory:login shell

- User information is stored in **/etc/passwd** file
  - account : login ID or username
  - password : encrypted field for the user password
    - `/etc/shadow` : contains password chunks
  - UID : user ID (UID), Linux identifies accounts with this ID.
  - GID : group ID (GID), ID of the default group of this account
  - GECOS : Optional field
    - General Electric Comprehensive Operating Systems
    - usually used for the full user name
  - home directory : the absolute path of the account
  - login shell : the default shell of the account
- **/etc/shadow** : Secure user account information
  - You can see the password, even though it is encrypted

# /etc/passwd

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
kbskim:x:1000:1000:Kyungbaek Kim,,,:/home/kbskim:/bin/bash
yez:x:1001:1001:Ye Zhao,,,,:/home/yez:/bin/bash
mglee:x:1002:1001::/home/mglee:/bin/bash
```

# Decoding user information

```
kbkim:x:1000:1000:Kyungbaek Kim,,,:/home/kbkim:/bin/bash
```

- Account : kbkim
- Password : x (not displayed to users)
- User ID : 1000
- Group ID : 1000
- GECOS (Optional Field): Kyungbaek Kim
- Home Directory : /home/kbkim
- Login shell : /bin/bash

# /etc/shadow

```
daemon*:16105:0:99999:7:::  
bin*:16105:0:99999:7:::  
peterpan:$1$4cxEeSCx$JeWhRsuySxowaR8mf5sKT0:16205:0:99999:7:::
```

Account : Password : Last changed : MIN : MAX : WARNING : INACTIVE : EXPIRE : R

- Account → Login name
- Password → Encrypted password
- Last changed → The date of the last password change, expressed as the number of days since Jan 1, 1970
- MIN → The minimum password age. The number of days the user will have to wait before she will be allowed to change her password again. The empty field and value 0 mean that there is no minimum password age.
- MAX → The maximum password age. The number of days after which the user will have to change her password.
- WARNING → The password warning period. The number of days before a password is going to expire. When the password is expired, no login is possible using the current user's password
- INACTIVE → The password inactive period. The number of days after a password has expired during which the password should still be accepted.
- EXPIRE → Account expiration date. After this date, the user shall not be allowed to login. (C.f. password expiration)

# UID

- User ID
  - A computer is a number-oriented machine.
    - Different accounts with the same UID are recognized as the same user of Linux
  - Regular user's UID usually starts from 1000
  - 0~999 and 65534 is assigned for Linux
    - 0 : UID of Root
    - 1 : daemon
    - 65534 : nobody

# Groups and GID

- Users may be grouped together into a “group”
- Users may choose to join an existing group to utilize the privileged access it grants
- All the groups on a system are listed in **/etc/group** file
  - Representing which users are included in which group
- A private GID for every UID of 1000 and greater is created
- GID of passwd file → the default group of the account
  - Other groups are described in /etc/group file

# /etc/group

```
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:kbkim,nmdo,yez
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
kbkim:x:1000:
dsm:x:1001:kbkim,mglee,yez
yez:x:1002:
```

Group name: x : GID : Group member

- Group name → name of the group
- x → group password, not shown to user
- GID → group id
- Group member → members of the group, separated by comma “,”

Kbkim

- Default group : kbkim (GID 1000)
- Supplementary groups : adm, dsm



# su command

- Use **su** to switch to a different user
  - Quicker than logging off and back on again
- Usually best to use **su** for working as root.

```
$ su - peter  
Password:
```

Changing to another user named peter

```
$ su -  
Password:
```

Changing to root

“-”, “-l”, or “--logging” ➔ Provide an environment similar to what the user would log in directly.

# who and whoami command

- who
  - Display who is on the system
- whoami
  - Display the effective username of the current user when invoked

```
$ whoami
kdkim
$ su -
Password:
# whoami
root
```

# Last command

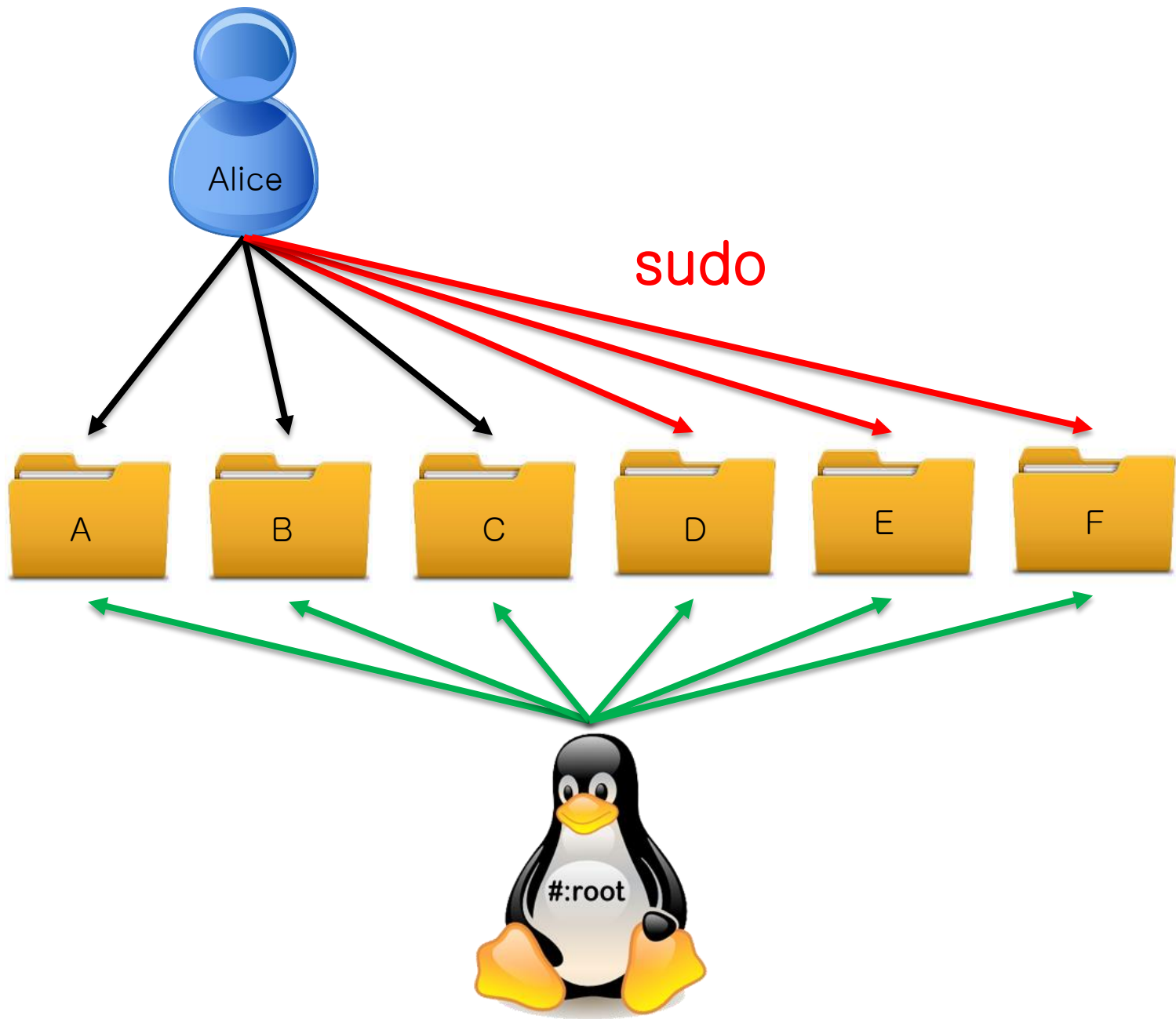
- last
  - Display account, login, logout, terminal or IP address

```
hduser@ubuntu: ~  
hduser@ubuntu:~$ last  
hduser pts/1 :0 Sun Mar 3 19:14 still logged in  
peterpan pts/1 :0 Sun Mar 3 19:11 - 19:14 (00:02)  
  
wtmp begins Sun Mar 3 19:11:12 2019  
hduser@ubuntu:~$
```

# sudo command

- Use **sudo** command to acquire root privilege without switching to root user
  - A user simply give his password to acquire root privilege through sudo
  - Once you give the password for sudo, you don't need to provide password again until its token is expired

```
$ whoami
kbkim
$ sudo whoami
[sudo] password for kbkim:
root
```



# sudoers

- Users who can perform “sudo”
  - So, we call sudoer ( “sudo” + “er” )
- */etc/sudoers* file controls sudoers

```
#User Specification Syntax → Account Host=Command  
root ALL=(ALL:ALL) ALL  
#user1 has root privilege while running “useradd”  
user1 ALL=/user/sbin/useradd  
  
#Members of group admin do not need a password  
%admin ALL=NOPASSWD: ALL  
#Members of the sudo group may gain root privileges  
%sudo ALL=(ALL:ALL) ALL  
#Add users to group “admin” or “sudo” to make them sudoers.
```

# Adding a user

- **useradd** command
  - Parameters for adding a new user
    - Username
    - `-m` : creating the user home directory (/home/[username])
    - `-g [default_group]` : defining the group name of the user's default login group
    - `-G [supplementary_groups]` : introducing a list of supplementary groups which the user is also a member; each group is separated by comma
    - `-p [password]` : defining the default password
      - Encrypted password : possibility of leak – not recommended option
    - `-d [home_directory]` : defining the home directory
    - `-s [login_shell]` : defining the path and filename of user's default login shell
    - `-o` : allow non-unique UID
  - e.g.) `useradd -m -g team1 steve`

# Adding a user : Example

- Make a user “stack” and make him a sudoer.

```
$ useradd -m -G admin -s /bin/bash stack
```

- Make a user “gslee” and set his default group to “faculty” group

```
$ useradd -m -g faculty -s /bin/bash gslee
```



# Checking default configuration of “useradd”

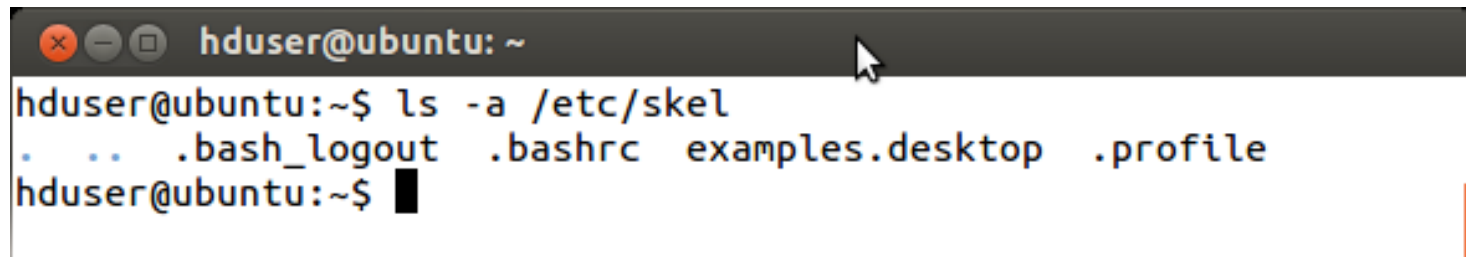
- useradd -D
  - Checking the default configuration of “useradd”
  - “/etc/default/useradd” contains the details of default configuration
    - Modify this file for the configuration

```
hduser@ubuntu: ~  
hduser@ubuntu:~$ useradd -D  
GROUP=100  
HOME=/home  
INACTIVE=-1  
EXPIRE=  
SHELL=/bin/sh  
SKEL=/etc/skel  
CREATE_MAIL_SPOOL=no  
hduser@ubuntu:~$
```

```
root@ubuntu: /etc/default  
# Default values for useradd(8)  
#  
# The SHELL variable specifies the default login shell on your  
# system.  
# Similar to DSHELL in adduser. However, we use "sh" here because  
# useradd is a low level utility and should be as general  
# as possible  
SHELL=/bin/sh  
#  
# The default group for users  
# 100=users on Debian systems  
# Same as USERS_GID in adduser  
# This argument is used when the -n flag is specified.  
# The default behavior (when -n and -g are not specified) is to create a  
# primary user group with the same name as the user being added to the  
# system.  
# GROUP=100  
#  
# The default home directory. Same as DHOME for adduser  
# HOME=/home  
#  
# The number of days after a password expires until the account  
# is permanently disabled  
"useradd" 37 lines, 1118 characters
```

# Resource skeleton for a user

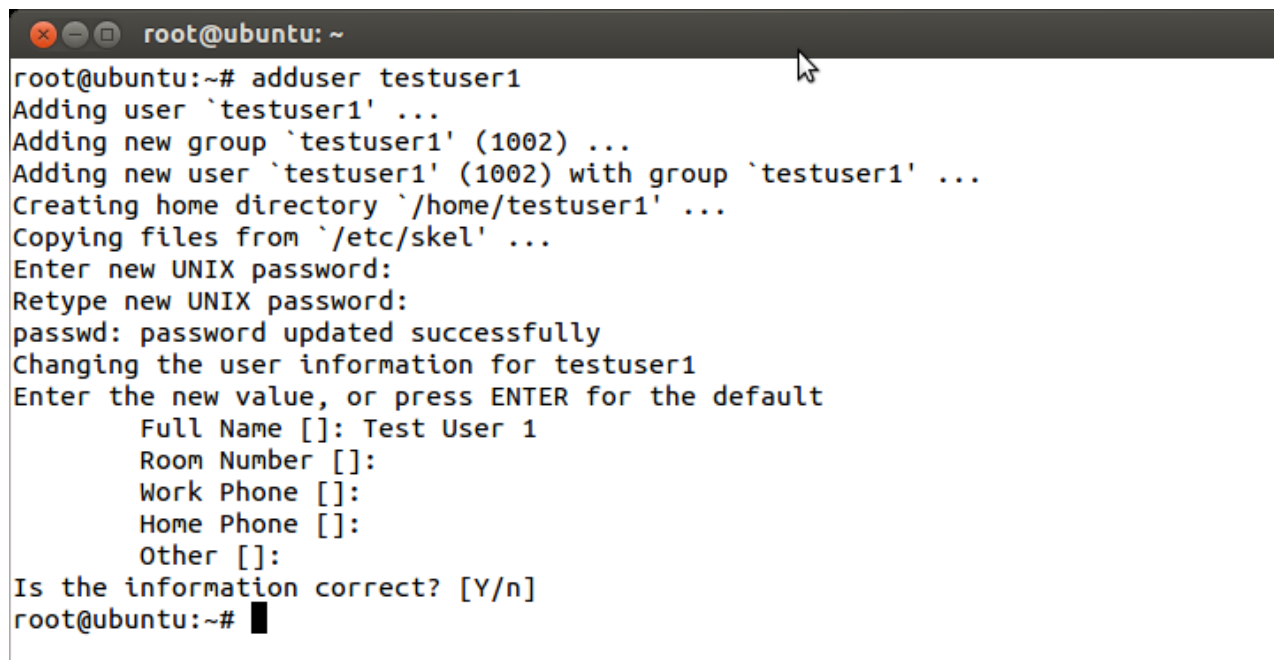
- “/etc/skel” directory
  - Contains the default files to be distributed when a new account is generated
  - Bash settings : “.bashrc”, “.bash\_logout”
  - Other files : “examples.desktop”

A terminal window with a dark title bar containing the text 'hduser@ubuntu: ~'. The terminal shows a command 'ls -a /etc/skel' being executed, resulting in the output: '. . .bash\_logout .bashrc examples.desktop .profile'. The prompt 'hduser@ubuntu:~\$' is visible at the end of the line.

```
hduser@ubuntu: ~  
hduser@ubuntu:~$ ls -a /etc/skel  
. . .bash_logout .bashrc examples.desktop .profile  
hduser@ubuntu:~$
```

# “adduser” command

- Alternative command for adding a user
- Options : `--uid UID`, `--gid GID`,  
`--home DIR`, `--shell SH`

A terminal window titled 'root@ubuntu: ~' showing the execution of the 'adduser testuser1' command. The output shows the creation of a new group 'testuser1' (1002), a new user 'testuser1' (1002) with that group, and the creation of a home directory. It prompts for a password, which is entered and confirmed. Finally, it prompts for user information (Full Name, Room Number, Work Phone, Home Phone, Other) and asks if the information is correct. The user enters 'Test User 1' for the full name and 'Y' for yes. The prompt returns to the root shell.

```
root@ubuntu:~# adduser testuser1
Adding user `testuser1' ...
Adding new group `testuser1' (1002) ...
Adding new user `testuser1' (1002) with group `testuser1' ...
Creating home directory `/home/testuser1' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for testuser1
Enter the new value, or press ENTER for the default
    Full Name []: Test User 1
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]
root@ubuntu:~#
```

# Modifying Users

- **usermod** command
  - Modifies the system account files to reflect the changes that are specified on the command line
  - e.g.) `usermod -g prof kbkim`
  - Options
    - Similar to the **useradd** command
    - c.f.) `-l` : change account name as a new name, should check the existence of home directory
    - c.f.) `-m` : move the user home directory (do not create the directory)

# Managing User

- **passwd** command
  - Specifying the password of a user
  - e.g.) passwd kbkim
- **chfn** command
  - Change the GECOS field
  - e.g.) chfn kbkim
- **userdel** command
  - Deleting a user account
  - -r option : removing home directories as well
    - e.g.) userdel -r kbkim
  - -f option : forcefully deleting a user account, even though the user is logging in

# Password aging related commands

- `passwd -n` → set the minimum password age
- `passwd -x` → set the maximum password age
- `passwd -w` → set the password warning period
- `usermod -f` → set the password inactive period
- `Usermod -e yyyy-mm-dd` → set account expiration date

# Managing Groups

- **groupadd** : creates and adds a new group
  - Without “-g” option, the next value of greatest GID will be assigned to a new group
  - E.g.) groupadd -g 1004 gradstudents
  - Alternative command : “addgroup”
- **groupmod** : changes name or GID
  - Options: -n name, -g GID
- **groupdel** : removes an existing group

# Managing Groups : check up

- **groups** command
  - Display group membership of a user
  - e.g.) groups kbkim
- **id** command
  - Display details of group information of user
  - UID and GIDs
  - e.g.) id kbkim

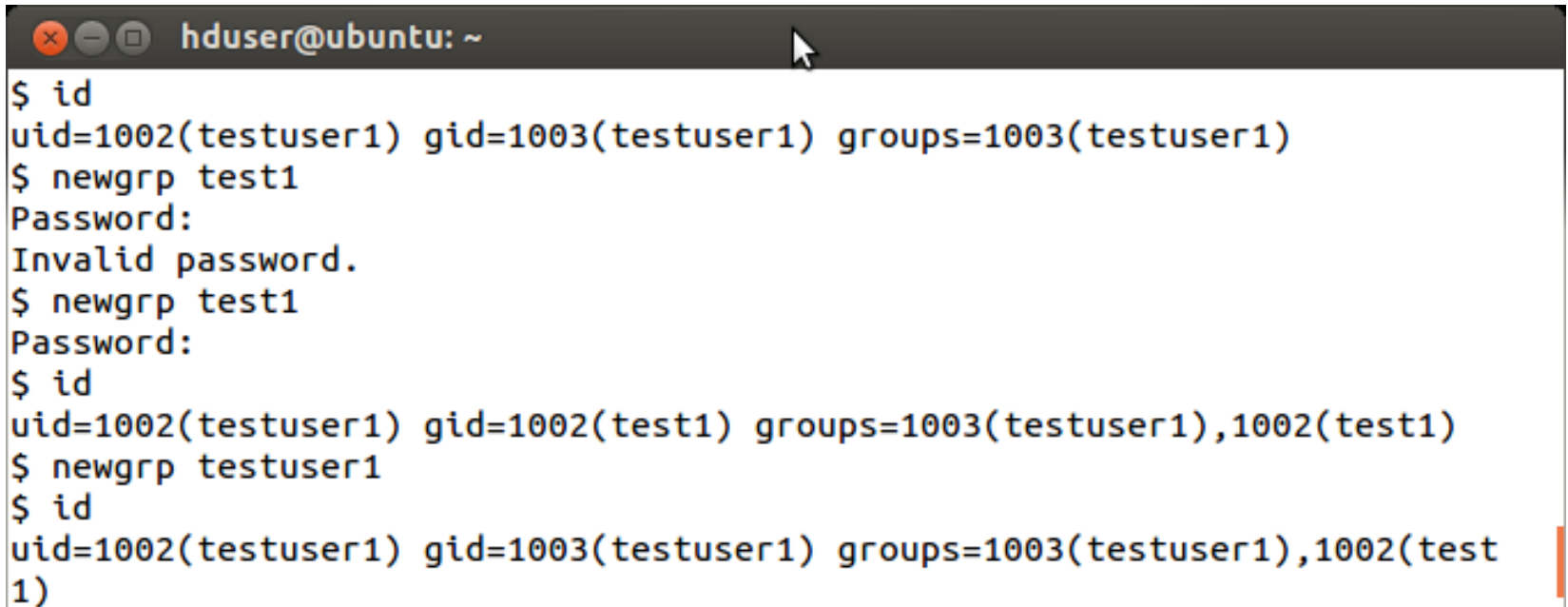


# Group password

- A group may have a password
- “gpasswd” command
  - Set the password of a group
  - Options
    - -a : add a user to a group
    - -d : delete a user from a group
    - -r : remove group password

# newgrp command

- Log in to a new group
  - Change the current group ID during a login session
    - Group password is required

A terminal window titled 'hduser@ubuntu: ~' showing the execution of the 'newgrp' command. The user first runs 'id', showing they are in group 1003. Then they run 'newgrp test1' and are prompted for a password, which is rejected as invalid. They run 'newgrp test1' again, are prompted for a password, and then run 'id' again, showing they are now in group 1002. Finally, they run 'newgrp testuser1' and 'id' again, showing they are back in group 1003.

```
$ id
uid=1002(testuser1) gid=1003(testuser1) groups=1003(testuser1)
$ newgrp test1
Password:
Invalid password.
$ newgrp test1
Password:
$ id
uid=1002(testuser1) gid=1002(test1) groups=1003(testuser1),1002(test1)
$ newgrp testuser1
$ id
uid=1002(testuser1) gid=1003(testuser1) groups=1003(testuser1),1002(test
1)
```