

Users and accounts

Chonnam National University
School of Electronics and Computer
Engineering

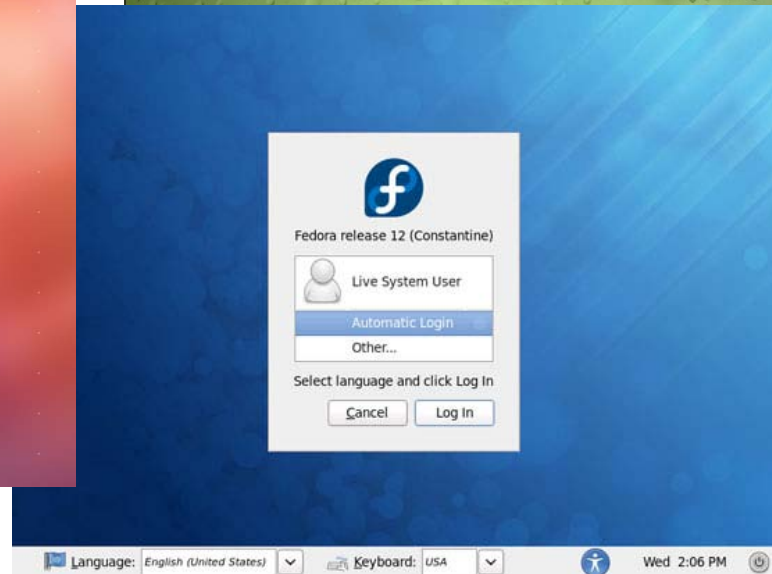
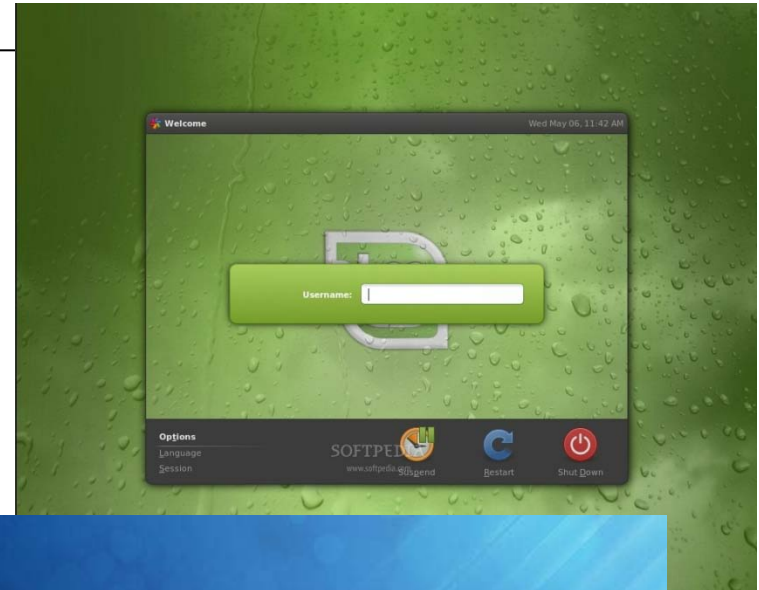
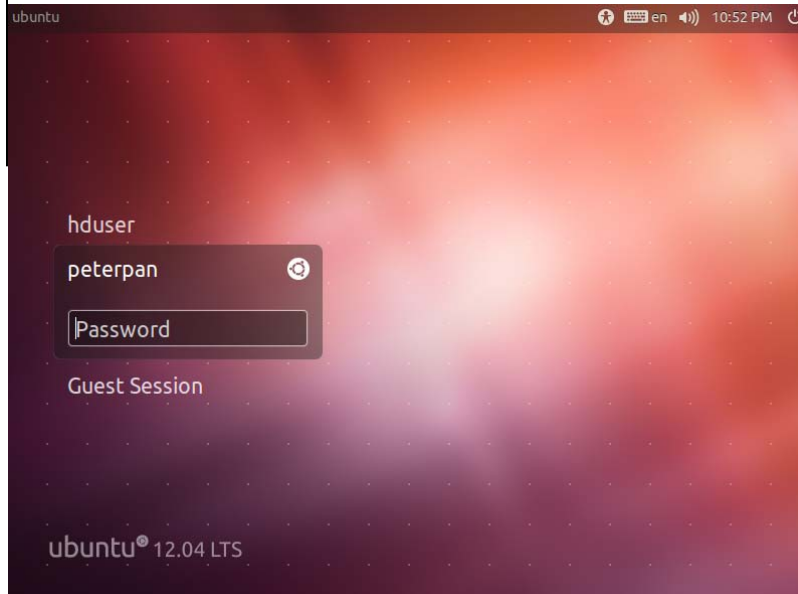
Kyungbaek Kim

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, logging in will present a **shell**
 - If graphical, logging in will present a desktop
 - A shell runs in the 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

- The shell is where commands are invoked
- A command is typed at a shell prompt
 - 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

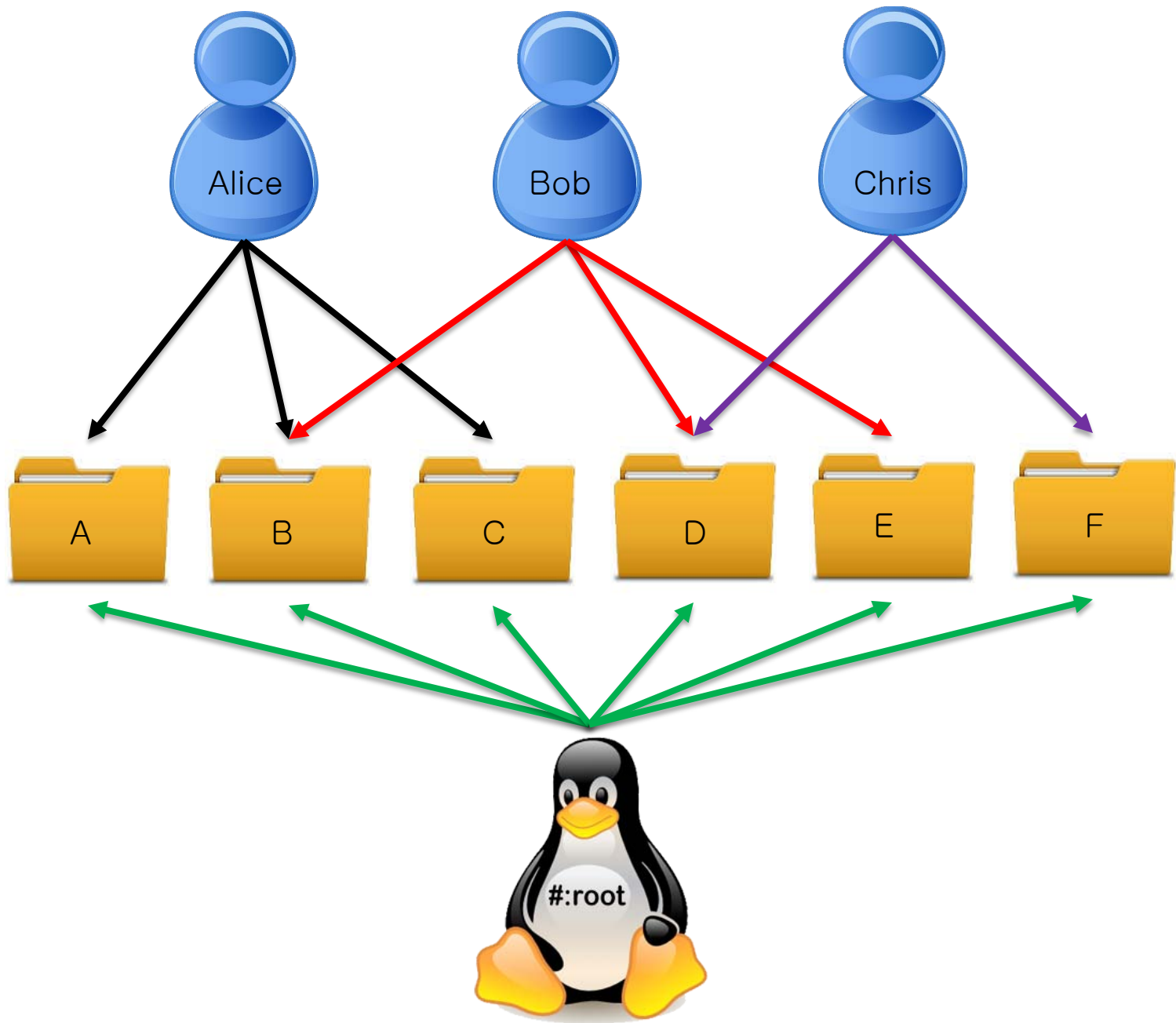
- To exit from the shell, use the **exit** command
- Pressing **Ctrl+D** at the shell prompt will also quit the shell
- Quitting all programs should log you out
 - In a text-only single-shell environment, exiting the shell should be sufficient
 - In a window environment, the window manager should have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

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 '#'



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 “-l” → Provide an environment similar to what the user would expect had the user logged in directly.

who and whoami command

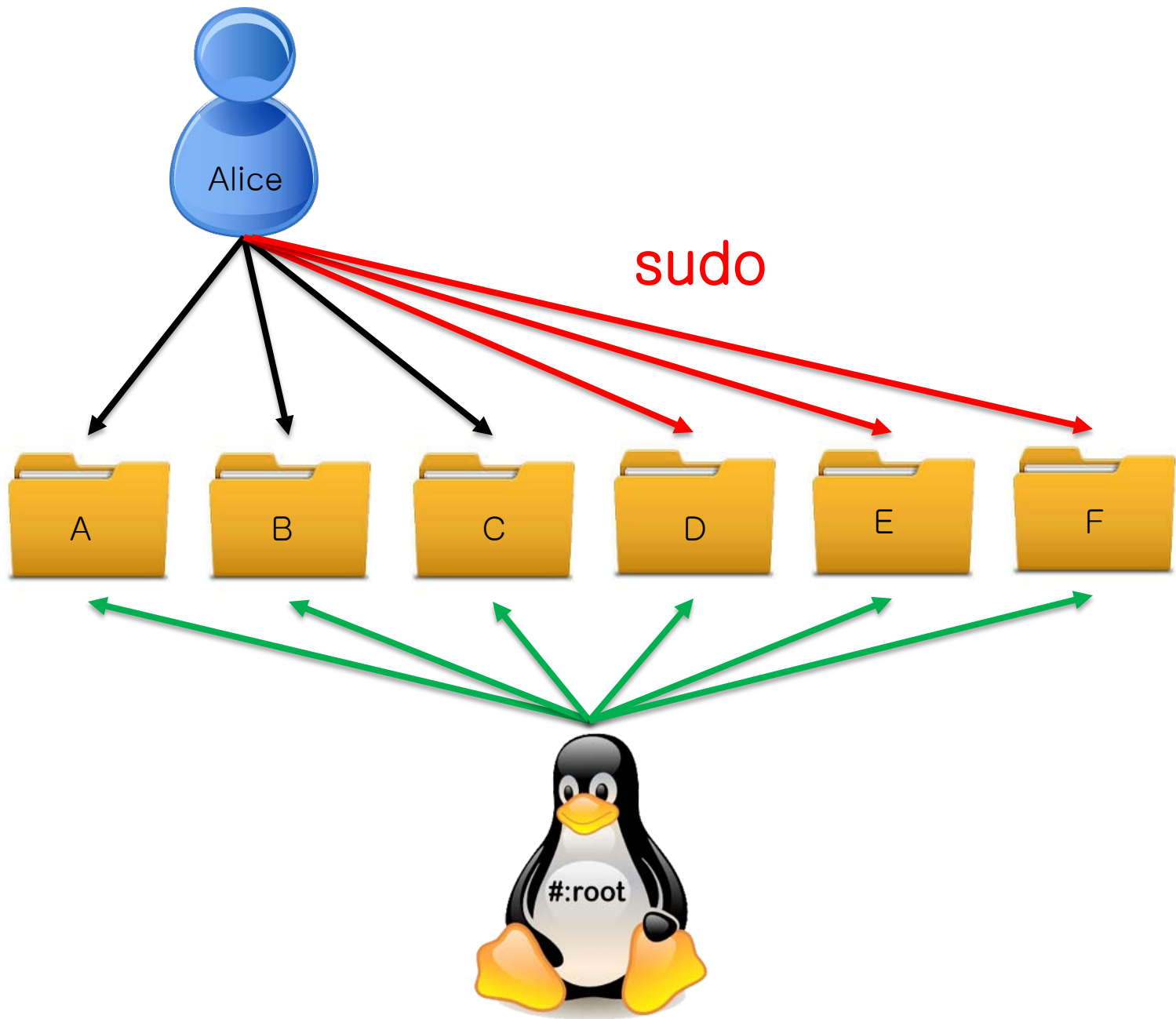
- who
 - Display who is on the system
- whoami
 - Display the effective username of the current user when invoked

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

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
```



User accounts

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

- User information is stored in **/etc/passwd** file
 - Username
 - encrypted field for the user **password**
 - user ID (**UID**)
 - group ID (**GID**)
 - GECOS–Optional field
 - General Electric Comprehensive Operating Systems
 - usually used for the full user name
 - home directory
 - default shell
- **/etc/shadow** : Secure user account information – you can see the password

Decoding user information

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

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

UID and GID

- User ID and Group ID
 - A computer is a number-oriented machine.
 - Root's UID is 0
 - Regular user's UID usually starts from 1000
 - A private GID for every UID of 1000 and greater is created

/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
```

Groups

- 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

/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:
```

sudoers

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

```
#Members of group sudo to not need a password
%sudo ALL=NOPASSWD: ALL
#Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
```

- Add users to group “admin” or “sudo” to make them sudoers.

Managing User : Adding

- **useradd** command
 - Parameters for adding a new user
 - Username
 - `-m` : creating the user home directory (`/home/[username]`)
 - `-g [initial_group]` : defining the group name of the user's initial login group
 - `-G [additional_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
 - `-s [login_shell]` : defining the path and filename of user's default login shell
 - e.g.) `useradd -m -g team1 steave`

Managing User

- **passwd** command
 - Specifying the user's password
 - e.g.) passwd kbkim
- **userdel** command
 - Deleting user account
 - -r option : removing home directories
 - e.g.) userdel -r kbkim
- **chfn** command
 - Change the GECOS field
 - e.g.) chfn kbkim

Managing User : modifying

- **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

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

Managing Groups : adding

- **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
- **groupdel** : removes an existing group
- **groupmod** : changes name or GID
 - Options: -n name, -g GID

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 initial group to “faculty” group

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