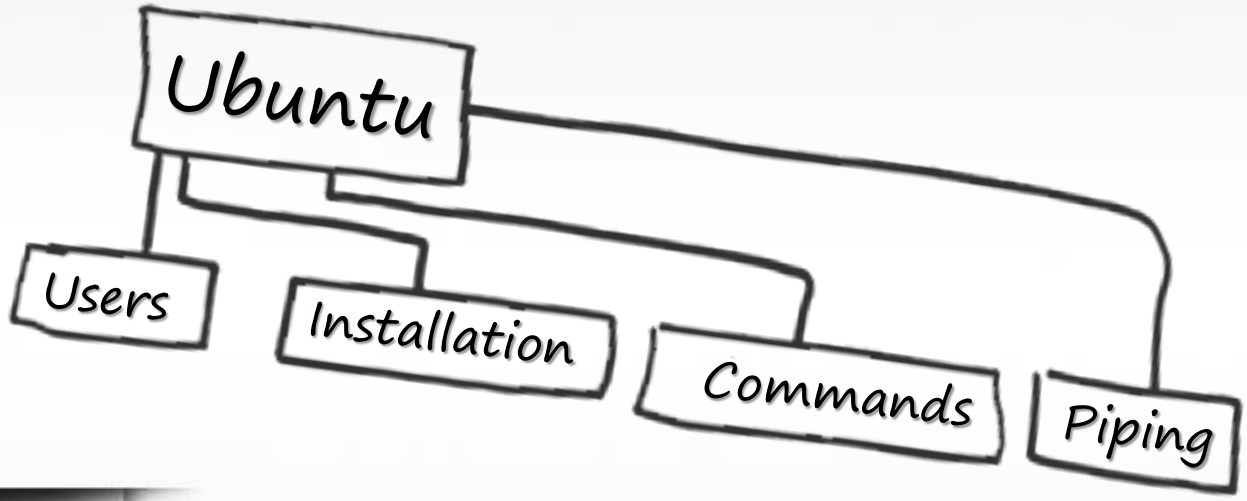
NOW

Ubuntu Fundamentals

ubuntu



OPEN SOURCE
DEPARTMENT

Information Technology Institute

Course Materials

You can access the course materials via this link

http://goo.gl/MZqU4b

Day 2 Contents

- Vi text editor
- Permissions and Advance Permissions
- User and group administration
- Switching to other accounts
- Shutting down the system

The vi text editor

- Default editor in all UNIX operating systems.
- Usually the only editor available in emergencies.
- Relatively hard to learn, but really powerful
- vi in Linux is usually vim (vi improved)
  - Syntax highlighting
  - Arrow keys, Del, BS work in insert mode
  - Mouse support

The vi text editor Cont'd

- VI editor is an interactive editor that you can use to create and modify test files.

- It is used when the desktop environment window system is not available.

Fundamentals VI Operations

- VI three basic modes
  - **Command mode**
    - Default mode
    - Perform commands to delete, copy, …

Fundamentals VI Operations

- VI three basic modes
  - **Edit mode**
    - Enter text into the file
  - **Last line mode**
    - Advanced editing commands
    - To access it, enter a colon (:) while in the command mode

Fundamentals VI Operations

- To enter edit mode
  - **i** Inserts text before the cursor
  - **o** Opens a new blank line below the cursor
  - **a** Appends text after the cursor
- After editing Press **esc** to enter command mode

Fundamentals VI Operations

- The syntax of vi command
  - **vi**
  - **vi** filename
  - **vi** options filename
- To recover a file
  - **vi –r** filename

Manipulating Files Within VI

- **Viewing files in Read-only mode**
  - view filename
    - Perform the **:q** command exit
- **Inserting and appending text**
  - **A** append text at the end of the line
  - **I** insert text at the beginning of the line
  - **O** opens a new line above the cursor

Manipulating Files Within VI

- **Moving the cursor within the vi**
  - **h**, left arrow, or backspace: left one character
  - **j** or down arrow: down one line
  - **k** or up arrow: up one line
  - **l**, right arrow or space: right one character

Manipulating Files Within VI

- **Moving the cursor within the vi (cont.)**
  - **w** forward one word
  - **b** back one word
  - **e** to the end of the current word
  - **0** to the beginning of the line
  - **Enter**: down to the beginning of the next line

Manipulating Files Within VI

- **Moving the cursor within the vi (cont.)**
  - **G** Goes to the last line of the file
  - **nG** Goes to Line n
  - **:n** Goes to Line n
  - **Control-F** Pages forward one screen
  - **Control-B** Pages back one screen
  - **Control-L** refresh the screen

Manipulating Files Within VI

- **Substitute and delete text**
  - **s** Substitutes a string for a character at the cursor.
  - **x** Deletes a character at the cursor.
  - **dw** Deletes a word or part of the word to the right of the cursor.
  - **dd** Deletes the line containing the cursor.
  - **D** Deletes the line from the cursor to the right end of the line.
  - **n,nd** Deletes Lines n through n

Manipulating Files Within VI

- **Search and replace**
  - **/string** Searches forward for the string.
  - **?string** Searches backward for the string.
  - **n** Searches for the next occurrence of the string.
  - **N** Searches for the previous occurrence of the string.
  - **%s/old/new/g** Searches for the old string and replaces it with the new string globally.

Manipulating Files Within VI

- **Copy and paste**
  - **yy** Yank a copy of a line.
  - **p** Put yanked text under the line containing the cursor.
  - **P** Put yanked text before the line containing the cursor.
  - **n,n co n** Copy Lines n though n and puts them after Line n.
  - **n,n m n** Move Lines n through n to Line n.

Manipulating Files Within VI

- **Save and quit**
  - **:w** save the file
  - **:w** new_file save as new file
  - :**wq, :x, ZZ** save and quit
  - **:q!** quit without saving

Manipulating Files Within VI

- **Customizing vi session**
  - **:set nu, :set nonu** show and hide line numbers
  - **:set ic, :set noic** ignore or be case sensitive
  - **:set showmode, :set noshowmode** display or turn off mode

Editing Files with gedit

- The gedit text editor is a graphical tool for editing text files.

- The gedit window is launched by selecting:
  - Search menu→ gedit

Users and Groups databases

- The `/etc/passwd` file

```
username:x:uid:gid:comment:home-directory:login-shell
```

Users and Groups databases

- The `/etc/shadow` file

```
username:encrypted passwd:last
changed:min:max:warn:??:expire:fu
ture-use
```

Users and Groups databases

- The `/etc/group` file

```
groupname:x:gid:comma-separated
list of group members
```

- The `/etc/gshadow` file ???

Adding a new user account


```
# useradd username
```

- The `useradd` command populates user home directories from the `/etc/skel` directory.

- To view and modify default setting

```
useradd -D
```

```
# passwd username
```

- Adding multiple user accounts

```
# newusers filename
```

Modifying user accounts

- To change a user's account information, you can:
  - Edit the `/etc/passwd` or `/etc/shadow` files manually
  - Use the `chage` command discussed later

Modifying user accounts

- To change a user's account information, you can:
  - Use the `usermod` command:
  - `usermod [options] username`
    - Useful options
      - To changes the login name use -l <login name>
      - To lock the password use -L
      - To unlock the password use -U

Deleting a user account

- To delete a user account you can
  - Manually remove the user from
    - `/etc/passwd` file
    - `/etc/shadow` file
    - `/etc/group` file
    - remove the user's home directory (`/home/username`)
    - and mail spool file (`/var/spool/mail/username`)
  - Use the `userdel` command.

```
# userdel [-r] username
```

Password Aging Policies

- The `chage` command sets up password aging

```
# chage [options] username
```

- Options
  - -m: to change the min number of days between password changes
  - -M: to change the max number of days between password changes
  - -E date: change the expiration date for the account
  - -W: change the number of days to start warning before a password change will be required

User private group scheme

- A traditional problem found in many UNIX/Linux environments is when administrators place all users in the same primary group. When users on such systems use a umask value of 002.

- Ubuntu solves this problem by assigning user a primary group for which they are the sole members.

- This "private" primary group has the same name as the user's username

Managing Groups

- Creating New Group

```
# groupadd groupname
```

- Modifying an Existing Group

```
# groupmod [options] groupname
```

- Deleting a Certain Group

```
# groupdel groupname
```

- List all file which are owned by groups not defined in `/etc/group` file

```
# find / -nogroup
```

Managing Groups cont'd

- You can use the `gpasswd` command to define
  - Group members
  - Group administrators
  - And to create or change group passwords

- Use the -r option to the `groupadd` command avoids using a GID within the range typically assigned to users and their private groups.

Changing Active Group

- To switch between groups you are member in, use `newgrp` command.

```
newgrp group
```

- To display the groups you are member in use `groups` command

```
groups
other root bin sys adm uucp mail
tty lp
```

Switching Accounts


```
# su [-] [username]
# su [-] [username] -c command
```

Using sudo command

- sudo is more secure
- sudo access is controlled by the `/etc/sudoers`.
  - This file is edited by visudo, an editor and syntax checker.
  - To give a specific group of users limited root privileges
    - User_Alias LIMITEDTRUST=st1,st2
    - Cmnd_Alias MINUMUM=/etc/init.d/httpd
    - Cmnd_Alias SHELLS=/bin/sh,/bin/bash
    - LIMITEDTRUST ALL=MININUM
    - user5 ALL=ALL,!SHELLS
    - %development station1=ALL, !SHELL

File Ownership and Permissions

- Every file and directory has both **user** and **group** ownership. A newly-created file will be owned by:
  - The user who creates it
  - That user's primary group (unless the file is created in a set group ID (SGID) directory; more on this file in the next lesson)

File Ownership and Permissions

- File ownership can be changed using `chown` command.
- Example

```
# chown user1 file1
# chown user1:group1 file1
# chown :group1 file1
```

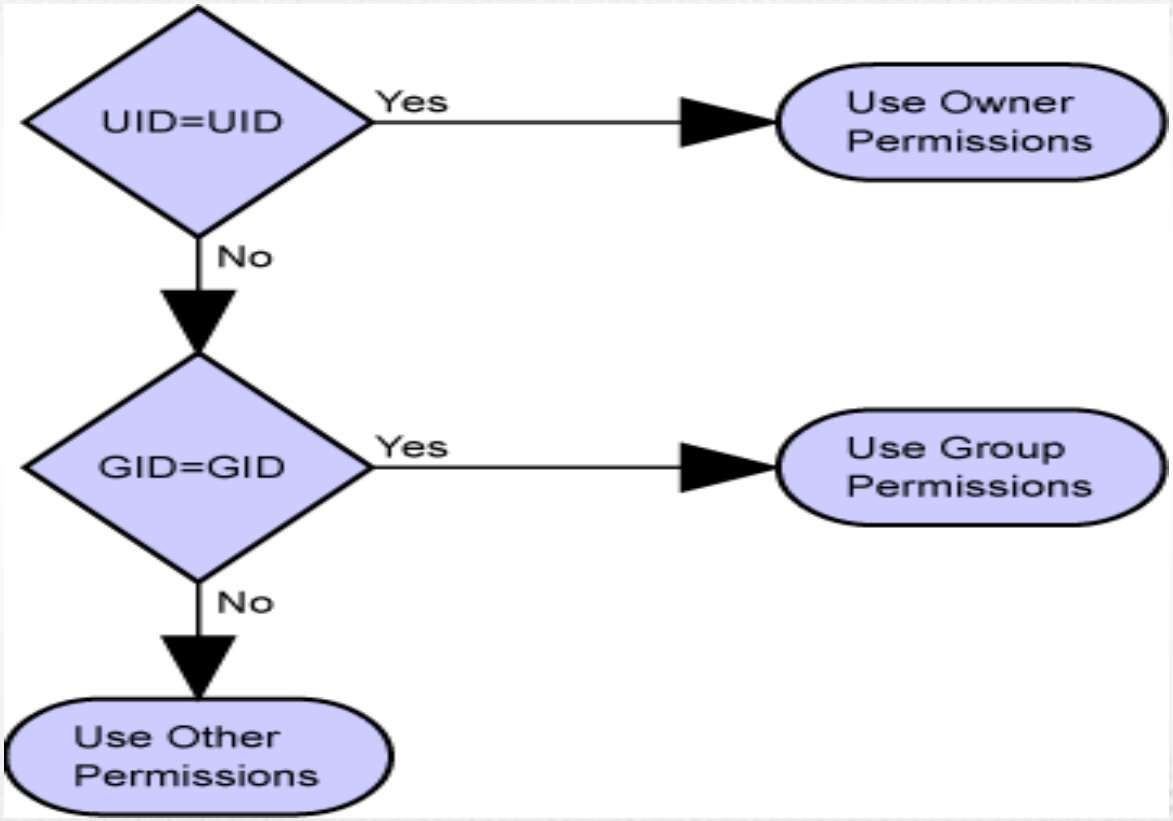
Security Scheme

- Each file has an owner and assigned to a group.
- Linux allows users to set permissions on files and directories to protect them.
- Permissions are assigned to
  - File owner
  - Members of the group the file assigned to
  - All other users
- The most specific permissions apply
- Permissions can only be changed by the owner and root

Permission Notations

Permission	Access for a File	Access for a Directory
Read	You can display file contents and copy the file.	You can list the directory contents with the ls command
Write	You can modify the file contents.	If you also have execute access, you can add and delete files in the directory.
Execute	You can execute the file if it is an executable. You can execute a shell script if you also have read and execute permissions.	You can use the cd command to access the directory. If you also have read access, you can run the ls -l command on the directory to list contents.

Determining Permissions



Changing the Permissions

`chmod permission filename`

- Permissions are specified in either
  - Symbolic mode
    - Who
      - u: Owner permissions
      - g: Group permissions
      - o: Other permissions
      - a: all permissions

Changing the Permissions

- Permissions are specified in either
  - Symbolic mode
    - Operator
      - + Add permissions
      - - Remove permissions
      - = Assign permissions absolutely
    - Permissions
      - r: read
      - w: write
      - x: execute

Changing the Permissions

- Permissions are specified in either
  - Octal mode
    - 4 read
    - 2 write
    - 1 execute

Examples

ls -l file1

```
-rw-r--r-- 1 user1 staff 1319 Mar 22 14:51 file1
```

chmod o-r file1

ls -l file1

```
-rw-r----- 1 user1 staff 1319 Mar 22 14:51 file1
```

chmod g-r file1

ls -l file1

```
-rw------- 1 user1 staff 1319 Mar 22 14:51 file1
```

Examples Cont'd

chmod u+x,go+r file1

ls -l file1

```
   -rwxr--r-- 1 user1 staff 1319 Mar 22 14:51
file1
```

chmod a=rw file1

ls -l file1

```
   -rw-rw-rw- 1 user1 staff 1319 Mar 22 14:51
file1
```

chmod 555 file1

ls -l file1

```
   -r-xr-xr-x 1 user1 staff 1319 Mar 22 14:51 file1
```

Examples Cont'd

chmod 775 file1

ls -l file1

```
-rwxrwxr-x 1 user1 staff 1319 Mar 22 14:51 file1
```

chmod 755 file1

ls -l file1

```
-rwxr-xr-x 1 user1 staff 1319 Mar 22 14:51 file1
```

Special Permissions

Special Permission	Effect on Files	Effect on Directories
u+s	File executes as the user that owns the file no the user that ran the file	No effect
g+s	File executes as the group that owns the file	Files newly created in the directory have their group owner set to match the group owner of the directory
o+t	No effect	Users with write permission on the directory can only remove files that they own

Special Permissions

- Example

```
# chmod u+s executable1
# ls -l
-rwsr-xr-x ...
# chmod g+s executable1
# ls -l
-rwxr-sr-x ...
# chmod g+s directory1
# ls -l
drwxr-sr-x ...
# chmod o+t directory1
# ls -l
drwxrwxrwt ...
```

Default Permissions

- The `umask` command sets the default permissions for files and directories

- Example
  - `# umask 002`
  - `# umask`
  - 022

Virtual Consoles

- Accessed with Ctrl-Alt-F_key
- Consoles 1-6 accept logins
- X server starts on the console 8

System Shutdown

- It only requires reboot or shutdown when you need to
  - Add or remove hardware
  - Upgrade to a new version of Ubuntu
  - Or upgrade your kernel
  - `shutdown -k now`
    - # doesn't really shutdown only send the warning messages and disable logins.
  - `shutdown -h time` # Halt after shutdown
  - `poweroff`
  - `init 0`

System reboot

- `shutdown -r`
- `reboot`
- `init 6`
- Press `CTRL+ALT+DEL`