

# Lecture – 12

## Getting Started with UNIX

References : Sumitabha Das

# SECTION -C

## Getting Started with UNIX

References : Sumitabha Das

# Introduction

- Logging in & logging out
- Format of UNIX commands
- UNIX Commands with explanation & purposes

## Logging in with Username & Password

- **Unix is security-conscious and can be used only by those persons who maintain an account with the computer system. The list of accounts is maintained separately in the computer.**
- **Because the system can be used by many users, someone has to be given charge of administration of the system. This person is known as the system administrator and he is the person who grant you the permission to use the system.**
- **He opens an account with a name for your use, and gives you a secret code that you have to enter when the system prompts you for it.**
- **If you are running UNIX on your desktop, then remember that you are the administrator of the machine.**

# Password

- The login prompt indicates that the terminal is available for someone to log in (i.e. .connect to the machine). This message also indicates that the previous user has logged out (i.e. finished her work and disconnected)
- Suppose you have an account named “kumar” , enter this string at the prompt. Then press *[Enter]* key after the string :

**login: kumar *[Enter]***

**Password: \*\*\*\*\* *[Enter]***

*Entry not*

*displayed*

- If you make mistake while typing, simply press *[Enter]* one or two times until the login prompt reappears on the screen. Be sure to terminate your responses with *[Enter]* to make system

- The string you entered at the prompt (login: ) is known as variously as your login name, user id, or user name.
- The secret code you entered at the next prompt (Password: ) is known as password.
- If you entered them incorrectly, the system flashes the following message:  
**Login incorrect**  
**login:**  
**Password:**
- Note: As soon as you log in, a program called the shell starts to run at your terminal, and keeps running until you terminate the session. As discussed before, the shell is your system's *command interpreter*. It displays the prompt and accepts all your input from the keyboard.

# Who is logged on?

The command who gives a list of logged on users:

**\$ who**

```
root          console          Jan   4 10:34
men6matw     tty1             Jan   6 09:45      (ecusun1)
cbl6nd              tty2             Jan   6 10:10
                (cblslcd)
cbl6ar       tty3             Jan   6 16:03      (cblsuna)
csc6ea              tty4             Jan   6 14:15
                (csuna1)
root          tty5             Jan   6 10:40
                (sun032)
ecl6rsh              tty6             Jan   6 15:39
csc6ea              tty8             Jan   6 14:15
                (csuna1)
```

# Logging out

- When you have finished your UNIX session you must log out from the system. To do this give the command:

**\$ logout**

You should always wait for the message confirming that you have logged out.

## Problems?

On some UNIX systems you may receive the message:

**logout: command not known**

If this happens you should type:

**exit**

You may occasionally get the message:

**There are stopped jobs**

If this happens simply give the logout command again.



# Groups

- The term *user* in UNIX is not meant to be only a person; it can represent a project or an application as well.
- A group of users performing similar functions may use the same username to use the system. It thus quite common to have usernames like marketing, accounts etc..
- When opening a user account, you have to associate the user with a group.
- A group usually has more than one member with a different set of privileges.
- People working on a common project should be able to read one another's files, which is possible only if they belong to the same group.

**Creating a user involves defining the following parameters:**

- **A user identification number (UID) and username.**
- **A group identification number (GID) and group name.**
- **The home directory.**
- **The login shell.**
- **The password**

## **groupadd : Command used for Adding a Group**

- **If the user is to be placed in a new group, an entry for the group has to be created first in /etc/group.**
- **A user always has one primary group and may also have one or more supplementary groups.**

- This file contains all of the named groups of the system, and a few lines of this file reveal the structure:

**root : x : 0 : root**

**bin : x : 1 : root, bin ,daemon**

**lp : x : 7 :**

**uucp : x : 14 : uucp, fax, root, fnet, sumit**

**users : x : 100 : henry, oracle, image, enquiry**

- Each line contains four colon-delimited fields. Lets focus our attention on the group named users shown in the first field.
- The second field once represented the group password but is hardly used today; it is either blank or an x.
- The third field shows the user's GID (here, 100). The last field contains a list of comma-delimited

- Note that primary group for a user is shown in `/etc/passwd`.
- To create a new group , dba, with a GID of 241 you have to use the `groupadd` command:

```
$ groupadd -g 241 dba
```

241 is the GID for

```
dba
```

The command places this entry in `/etc/group` which you can also insert manually.

```
dba : x : 241 :
```

# passwd : Changing your password:

If your account doesn't have a password or has one that is already known to others, you should change it immediately.

This is done with the passwd command :

**\$ passwd**

**passwd: Changing password for kumar**

**Enter login password : \*\*\*\*\***

*Asks for old password*

**New password : \*\*\*\*\***

**Re-enter new password: \*\*\*\*\***

**passwd (SYSTEM): passwd successfully changes for kumar.**

passwd (note the spelling) expects you to respond three times.

- (1) It prompts for the old password.
- (2) It checks whether you have entered valid password, and if you have, it then prompts for the new password. Enter the new password using the password naming rules applicable to your system.
- (3) Finally passwd asks you to re-enter the new password. If everything goes smoothly, the new password is registered by the system.

## Password framing rules:

- Depending on the way they are configured, many systems conduct certain checks on the string that you enter as password.
- There is often a minimum length of the string. Many systems insist on using a mix of letters with numerals.
- They may either disallow you from framing easy-to-remember passwords or advice you against choosing a bad password.

### The following messages are quite common:

**passwd (SYSTEM): Password too short – must be at least 6 characters.**

**passwd (SYSTEM): Passwords must differ by at least 3 positions.**

**passwd (SYSTEM): The first 6 characters of the password must contain at least two alphabetic characters and at least one numeric or special character.**

**BAD PASSWORD: it is based on a directory word.**

**BAD PASSWORD: is too similar to the old one.**

**BAD PASSWORD: Too many failures, try later.**

**These messages suggest that you are not able to choose any**



**These are some of the rules that you are expected to follow when handling your own password :**

- **Don't choose a password similar to the old one.**
- **Don't use commonly used names like names of friends, pets and so forth. A system may check with its own dictionary and throw out those passwords that are easily guessed.**
- **Use a mix of alphabetic or numeric characters. Enterprise UNIX systems won't allow passwords that are wholly alphabetic or numeric.**
- **Don't write down the password in an easily accessible**

## Format of UNIX Commands

- Unix commands have the general format:

**command** [*options*] [*item*]

- Items in brackets are optional, and words in italics are generic identifiers (i.e. *options* must be replaced by a particular option, eg. -a)
- Note that :  
Commands are case sensitive. The command **ls** is different from **LS**. In fact LS is not recognized as a valid command.
- Command options consist of a single character. The command to list all the files in a directory is **ls -a** and could **not** be **ls -all** (the latter would have to mean a combination of options.)

- **Command options can usually be combined or listed separately. For example:**
- **\$ ls -al or \$ ls -a -l** *This command lists out the files with dates and sizes.*
  - rw-r--r-- 1 david user 2046 Apr 4 22:29 .README**
  - rw-r--r-- 1 david user 1262 Jul 10 21:06 .cshrc**
- **As you look at the above file listings, you will see the following columns: access permissions, owner, group, size, modification date, name For example, check the listings for .README:**
  - **Access:** -rw-r--r--
  - **Owner:** david
  - **Group:** user
  - **Size:** 2046
  - **Modification Date:** Apr 4 22:29
  - **Name:** .README
- **The command item is given last. This is very often a file name. For example:**

- **bc** obtained the expressions from redirected standard input, processes them and sent out the results to a redirected output stream. Though we used the <first and then the >, there's no restriction in the sequence we use:

```
bc > result.txt <calc.txt
```

```
>result.txt < calc. txt bc
```

*same as above but command at end*

- The indifference of a command to the source of its input and destination to its output is one of the most profound features of the UNIX system.
- It raises the possibility of commands talking to one another so the output of one command can be used as input to another.

# Applications in Games

There are lots of fun things and games you can use in UNIX. Most of the ones listed below are local to Brown University. Try each of these commands. Check the man pages or the links below if you have trouble, but note, some of the commands do not have man pages. Have fun!

- [banner](#)
- [figlet](#)
- **WhatsForDinner**
- **food dilbert**
- [Forecast](#)
- **fortune say**
- [Xteddy](#)
- **xdeady**
- [BattleTris](#)
- [nethack](#)
- **Netris**
- **xbill**
- **xblast**
- [xboing](#)
- **xroach**

# Research

- **Unix Commands** CCR's computing resources are primarily Linux based and therefore using them requires a basic understanding of the Unix operating system. Some basic commands are provided below.
- **Basic Unix Commands** CCR Reference Card for Linux/UNIX commands [pdf](#)
- Show pathname of current directory: pwd
- List files: ls
- Make a directory: mkdir directory-name
- Change directory: cd directory-name
  - Change directory back to home directory: cd
- Copy a file: cp old-filename new-filename
- View a file:
  - cat filename
  - more filename
  - less filename
- Edit a file:
  - emacs filename
  - vi filename
- Delete a file: rm filename
  - Delete a directory (recursively): rm -R directory-name
    - All files and subdirectories are deleted
- Move a file: mv old-filename new-filename
- Change permissions:
  - Arguments to chmod command: ugo+-rwx
    - where ugo are user, group and other; rwx are read, write and execute
  - Add execute permission for yourself: chmod u+x filename
  - Remove read, write and execute for group and other from a directory its contents:  
chmod -R go-rwx directory-name