PRONET

VNU-ITP

UNDP

Chapter 5

Command-line

Objectives

PRONET

VNU-ITP

UNDP

- Define command line basic procedures
- List common commands
- Describe and use the commands `su` and `sudo`
- Describe and perform basic file system tasks
- Describe the Linux file system, its structure, and directory hierarchy

Objectives

PRONET

VNU-ITP

UNDP

- Locate commands and files
- Identify Linux text editors (e.g `vi`)
- Describe and use the Linux shell (e.g `bash`) environments
- Describe and use text processing tools and filters

Command Line Basic Syntax

PRONET VNU-ITP UNDP

- Usually, there are *three* components in command line syntax :

```
<Command> [Options] [Arguments]
```

<Command> : *what* the system will do

<Options> : or <switches>, *how* the command will do

<Arguments> : *where* the command will apply to

- Sometimes you won't need option or argument, depending on the command

Command Line Examples

PRONET VNU-ITP UNDP

- For example:

`$ date`	(Command)
`$ date mmddhhmmyy`	(Command and argument)
`$ cal 12 2000`	(Command and two arguments)
`$ uname -a`	(Command and option)
`$ uname -rpns`	(Command and multiple options)
`$ uname -r -p -n -s`	(Command and multiple options

COMMON COMMANDS

PRONET
VNU-ITP
UNDP

- Command line utilities are an extremely powerful way to complete day-to-day activities, you should be familiar with it
- Remember : Linux (Unix) is *case-sensitive*, “ls” , “LS” are different commands
- More information about a command can be found using `man` pages.

COMMON COMMANDS

PRONET VNU-ITP UNDP

`pwd`	Displays the current working directory
`cd`	Change working directory
`ls`	List contents of directories
`cp`	Copy files and directories
`mv`	Move or rename files
`rm`	Remove (delete) files or directories
`find`	Search for files in a directory hierarchy
`more`	Displays a file one page at a time
`grep`	Print lines matching a pattern
`file`	Determine file type

su And sudo

- Working as *root* can be risky business, it is usually better to work as the root user on a temporary basis
  - `su –c "command"` :This will *perform the command with root permission and then return to the normal user shell*. You will be prompted for the root password before the command is executed
    - **Ex.** *su –c "more /etc/shadow"*
  - `sudo`: This command allows user (as specified in the `/etc/sudoers` file) to execute some root commands

FILE SYSTEM BASICS

PRONET
VNU-ITP
UNDP

- File systems are *used to store files in an organized structure*

- It is important to have an understanding of how to *navigate* it and how to *create*, *move*, *copy* and *remove* files as a user

File System Structure

PRONET
VNU-ITP
UNDP

- Linux file systems have the following structural parts: the *superblock*, *inode* blocks and the *data* blocks
- The *superblock* contains *information about the whole file system*
- The *data* block are *a segment of disk space* in which the file contents is stored
- An inode number is assigned to each file that is created. An *inode itself contains all information about a file except its name*. The names of files are kept in directories , which are just *special files containing a table with file names and their inodes*

Directory Hierarchy

PRONET
VNU-ITP
UNDP

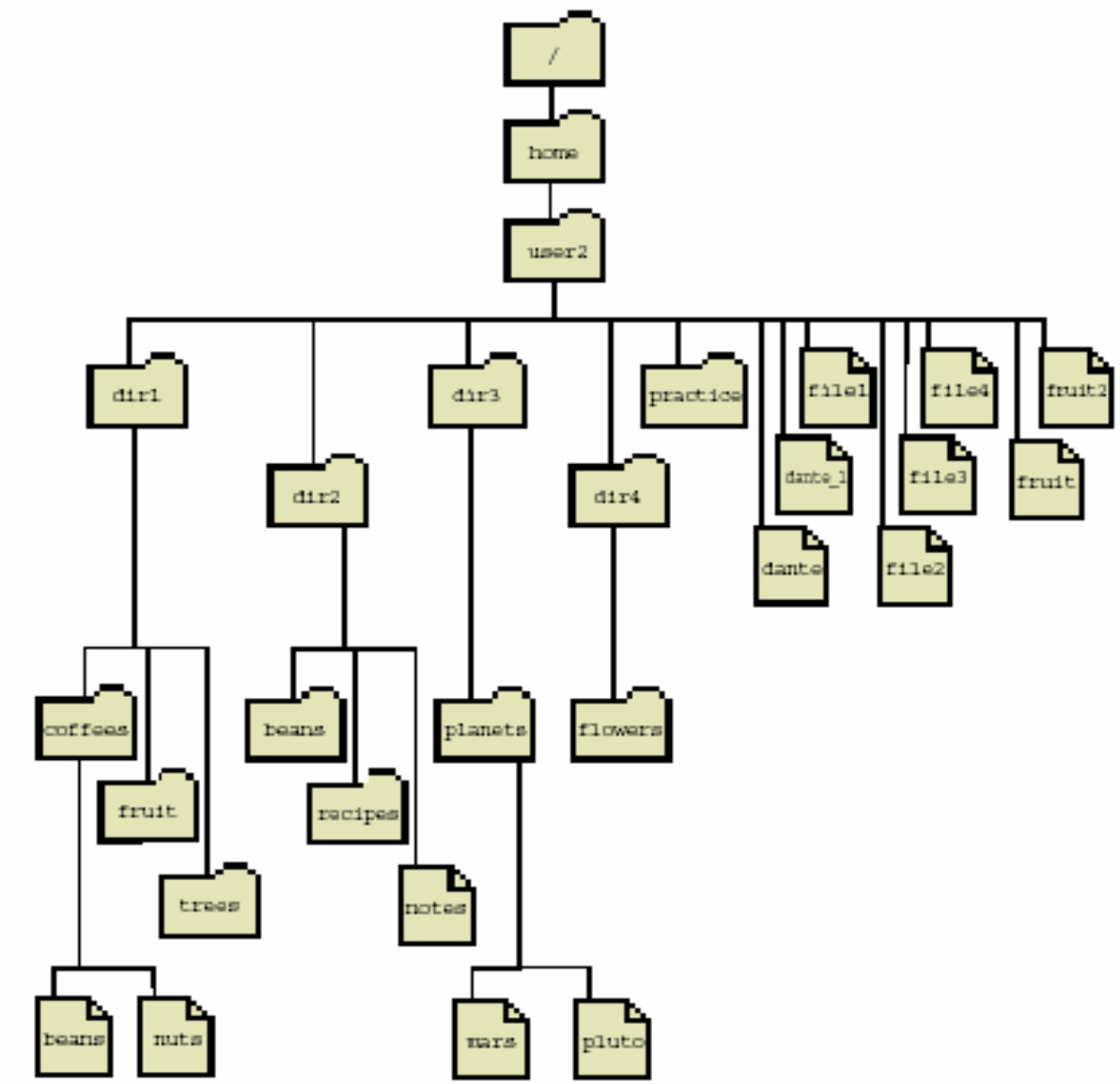
- The Linux file system hierarchy has a tree of directories.
- The tree starts with the "root" directory, "/"; from here the tree branches out into sub-directories and sub-sub-directories.
- Files are located inside some directories (like a leaf at the end of a branch)

Directory Hierarchy

PRONET

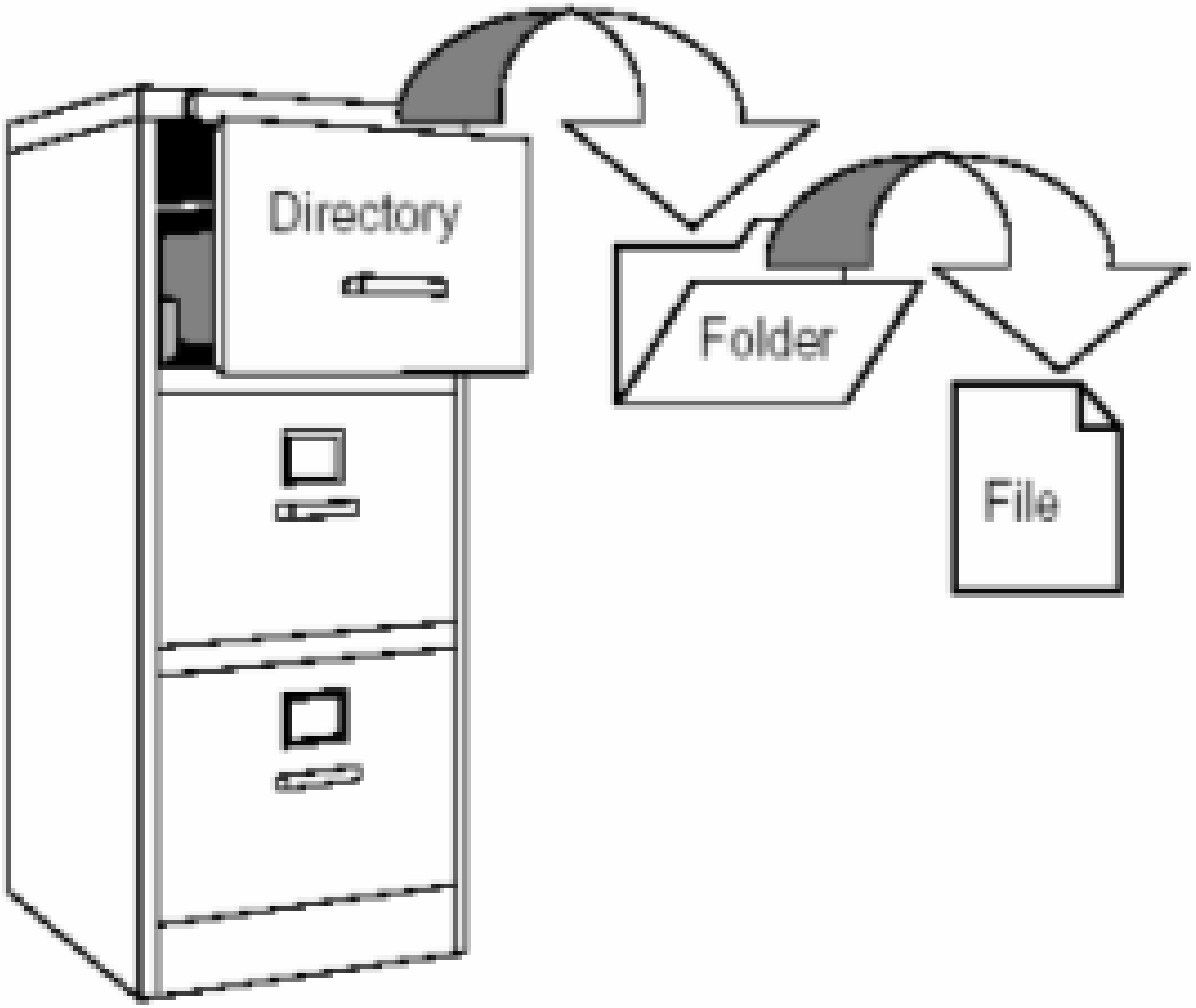
VNU-ITP

UNDP



File System Organization Example

PRONET
VNU-ITP
UNDP



Path Names

PRONET
VNU-ITP
UNDP

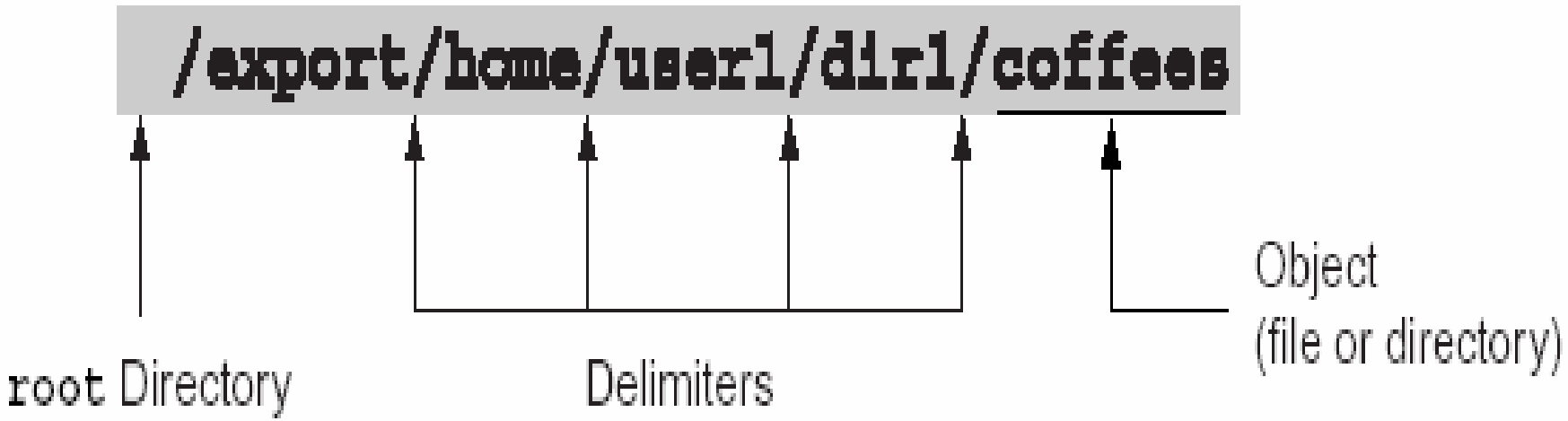
- Uniquely identify a particular file or directory by specifying its location in the directory tree.
- The slashes (/) within a path name are delimiters between object names.
- The first slash in a path name always represents the root (/) directory.

Path Names

PRONET

VNU-ITP

UNDP



Absolute Path Name

PRONET

VNU-ITP

UNDP

- An *absolute path name* specifies a file or directory in relation to the entire directory tree
- Absolute path names always:

  - Start at the root (/) directory, and then list each directory along the path to the final destination

  - Use a slash (/) to separate multiple directory or file names

Relative Path Name

PRONET
VNU-ITP
UNDP

- A *relative path name* describes the location of a directory or file as it relates to the current directory
- Relative path names:

  - Never begin with a slash (/) character

  - Use slashes (/) within the path name as delimiters between object names

File and Directories Naming

PRONET
VNU-ITP
UNDP

- Directory and file names can be up to 255 alphanumeric characters in length.
- Non-alphanumeric characters are allowed in file and directory names.
- Special characters should not be used.
- Spaces should not be used.
- Names do not need to contain extensions, but may if desired
- File name starts with a dot '.' is a *hidden* file

Pathname Abbreviations

PRONET VNU-ITP UNDP

.	Current (working) directory
. .	Parent directory; the directory directly above the current directory
~	User's home directory
`~username`	The home directory of the user specified by username.

Changing Directories

PRONET
VNU-ITP
UNDP

- When you initially log in to the system, the current directory is set to your home directory. Change your current working directory at any time by using the **`cd`** command.

- Command format:

```
cd directory_name
```

Displaying the Current Directory

PRONET
VNU-ITP
UNDP

- Use the **pwd** command to identify in which directory you are currently working.
- The **pwd** command displays the absolute path name of the current working directory.
- Command format: `pwd`

Displaying the Contents of a Directory

PRONET
VNU-ITP
UNDP

- Use the **`ls`** command to list the files and directories within the specified directory.
- Using the **`ls`** command with no argument displays the contents of the current directory.
- Command format:

```
ls [options] pathname...
```

Some ls Command Options

PRONET
VNU-ITP
UNDP

-a: List all entries, including those beginning with a dot (hidden files, except ".", "..")

-d: If argument is a directory, then do not list the contents of that directory

-l: List in a long format

-F : Display file type (/, *, @)

-R: Recursively list the contents of all subdirs.

Creating Directories

PRONET
VNU-ITP
UNDP

- Use the `mkdir` command to create directories

- Command format:

```
mkdir directory_name...

mkdir [-p] directory_name...
```

Copying Files

PRONET VNU-ITP UNDP

- Use the `cp` command to copy files and directories.

- Command format:

```
cp [-i] source_file destination_file
cp [-i] source_file(s) destination_directory
```

Copying Directories

PRONET
VNU-ITP
UNDP

- Use the `cp -r` command to copy a directory and its contents to another directory.

- Command format:

```
cp -ir source_directory(s)
            destination_directory
```

Moving and Renaming

PRONET

VNU-ITP

UNDP

- Use the `mv` command to move or rename a file or directory.

- Command format:

```
mv [-i] source target_file

mv [-i] source(s) target_directory
```

Removing Files

PRONET
VNU-ITP
UNDP

- Use the `rm` command to remove unwanted files and directories. Remove a single file or several files at once.

- Command format:

```
rm [-i] filename...
```

Removing Directories

PRONET
VNU-ITP
UNDP

- Remove unwanted directories by using the `rmdir` and `rm` commands.
  - The `rmdir` command deletes **empty** directories only.
  - The `rm -r` command removes a directory that contains files.

- Command format:

```
rmdir directory_name(s)

rm -r[i] directory_name(s)
```

LINUX TEXT EDITORS

PRONET
VNU-ITP
UNDP

- Mastering a text editor is an essential component of becoming proficient at Linux system administration.
- There are various text editors available with Linux :
  - vi
  - emacs and xemacs
  - jed
  - joe
  - ...
- We will cover `vi` , due to the fact that nearly every distribution of Linux and UNIX includes `vi` by default

Introducing the vi Editor

PRONET
VNU-ITP
UNDP

- The visual display (***vi***) editor is an interactive editor used to create or modify text files.
- All text editing with ***vi*** editor takes place in a buffer. Changes can either be written to disk or be discarded.
- The ***vi*** editor uses a nominal amount of system resources.
- The ***vi*** editor does not depend upon a windowing system.
- The ***vi*** editor is present on all UNIX systems.

vi Modes

PRONET
VNU-ITP
UNDP

- Command mode
- Edit mode
- Last line mode (ex mode)

Command Mode

PRONET VNU-ITP UNDP

- This is the default mode for the ***vi*** editor.
- You can enter commands to delete, change, copy, and move text; position the cursor; search for text strings; or exit the ***vi*** editor.

Edit Mode

PRONET
VNU-ITP
UNDP

- In this mode, you can enter text into a file.

- To instruct the ***vi*** editor to enter edit mode, enter one of the following three commands:

  i (insert)

  o (open)

  a (append)

Last Line Mode

PRONET
VNU-ITP
UNDP

- While in command mode, you can use advanced editing commands by typing a colon (:), which places you at the bottom line of the screen.

Switching Between *vi* Modes

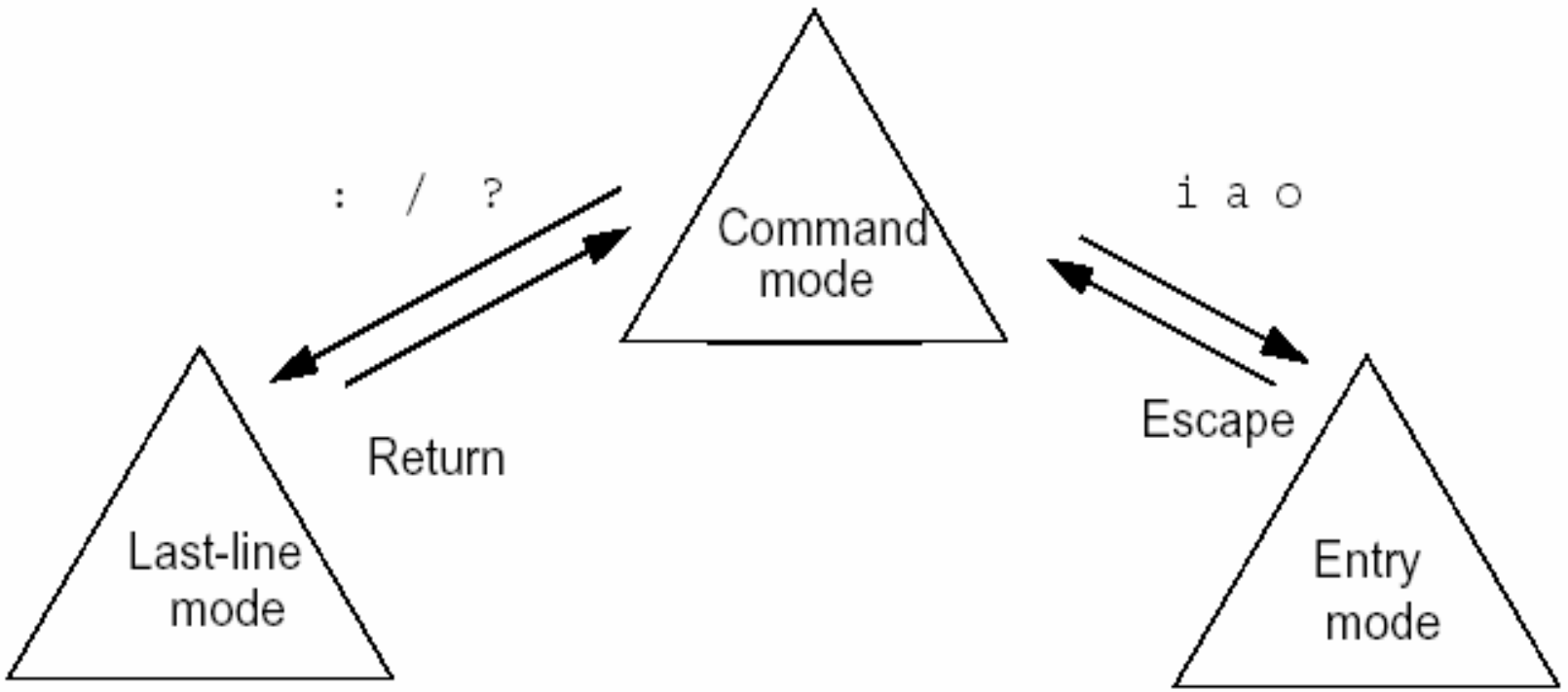


Figure 9-1 vi Modes

PRONET
VNU-ITP
UNDP

Invoking the *vi* Editor

PRONET VNU-ITP UNDP

- To create a new file, invoke the ***vi*** editor with a new file name by typing commands to create, edit, or view a file.

- Command format:

```
vi [option(s)] [filename]
view filename
```

Exiting the *vi* Editor

PRONET

VNU-ITP

UNDP

Command	Meaning
`:w`	Save changes (write buffer)
`:w new_filename`	Write buffer to `new_filename`
`:wq`	Save changes and quit `vi`
`ZZ`	Save changes and quit `vi`
`:q!`	Quit without saving changes
`:wq!`	Save changes and quit `vi` (The ! will override read only permissions if you are the owner of the file.)

THE LINUX SHELL

PRONET
VNU-ITP
UNDP

- *Shell* is basically an interactive environment from which a user communicates with the operating system
- There are several shells available: C shell (csh), the Korn shell (ksh), and the Bourne shell (sh), BASH shell (bash), …
- BASH (Bourne Again Shell) is a default shell on most Linux distributions

THE LINUX SHELL

PRONET
VNU-ITP
UNDP

- When a program is started from your typing at command line, a new shell is started running the program. It will have a seperate *environment*

Profiles

PRONET
VNU-ITP
UNDP

- When `bash` is invoked as an interactive login shell, it reads the following profiles:
  - `/etc/profile` :*system-wide* settings
  - `~/.bash_profile, ~/.bash_login, ~/.profile` : *user* settings. It reads and executes commands from the first one that exist and readable

Environment

PRONET
VNU-ITP
UNDP

- Environment is all settings that help program's running correctly. Most current settings are displayed by the (built-in) command `set` :
  - Settings of customizable parameters built into the shell
  - Environment variables *(envars)* and local shell variables

Environment variables

PRONET
VNU-ITP
UNDP

`PATH`	List of directories to search for commands, separated by colon
`PS1`	Shell prompt
`DISPLAY`	Current X display
`PAGER`	Preferred viewer (more, less etc.)
`PWD`	Current working directory
`SHELL`	Current shell (usually /bin/bash)
`HOME`	User's home directory
`HOSTNAME`	Name of the computer

Using the `bash` Shell

- One extremely convenient feature of Linux is the ability to switch between many shells, or virtual consoles (VC).
- If `gpm` is running, text can be copied and pasted from one console to another using the mouse
- Use the up and down arrow keys, you can scroll through a list of commands that previously executed from the command line (stored in ~/.bash_history)

Using the bash Shell

PRONET VNU-ITP UNDP

- *wildcard* characters :

  * matchs any of *zero* or *more* characters

  ? matchs any *single* characters

Example :

```
# ls image?.jp*g
image1.jpg
image3.jpeg
( but do NOT list image20.jpg file)
```

Command History and Editing

PRONET
VNU-ITP
UNDP

- *History* is a mechanism that allows user can get previous executed commands. Then you can *edit* it before executing.
- Executed commands will be saved in ~/.bash_history when you exit the shell. During session, they are temporary stored in buffer
- Each command has a corresponding number in history file

Command History and Editing

PRONET

VNU-ITP

UNDP

- List of useful command work with *history* :

!<*n*>	Reexecutes command *n* from history
!-<*n*>	Reexecutes the current command minus *n*
!<*string*>	Reexecutes the most recent command *starting* with *string*
^<*string1*>^<*string2*>	Repeats the last command but replaces *string1* with *string2*
!?<*string*>	Reexecutes the most recent command containing *string*

Recursive Commands

PRONET

VNU-ITP

UNDP

- Some commands, such as `rm`, `ls`, ... use a switch `-r`, `-R`, or `--recursive` to indicate that the commands is to be executed recursively through directories

Redirection

PRONET

VNU-ITP

UNDP

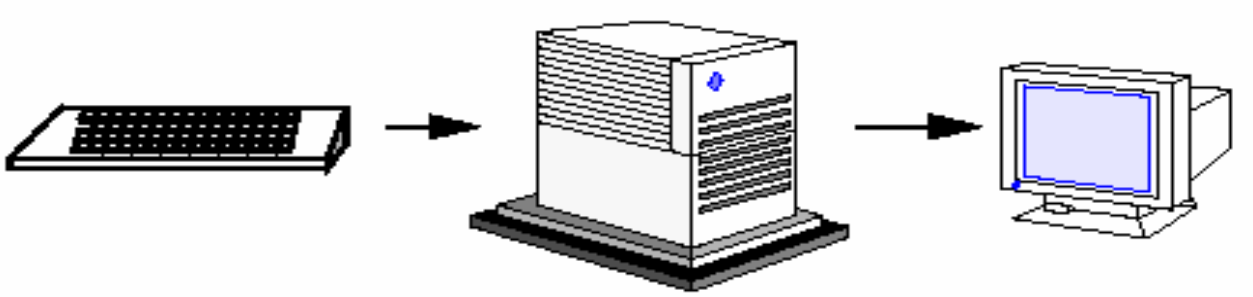
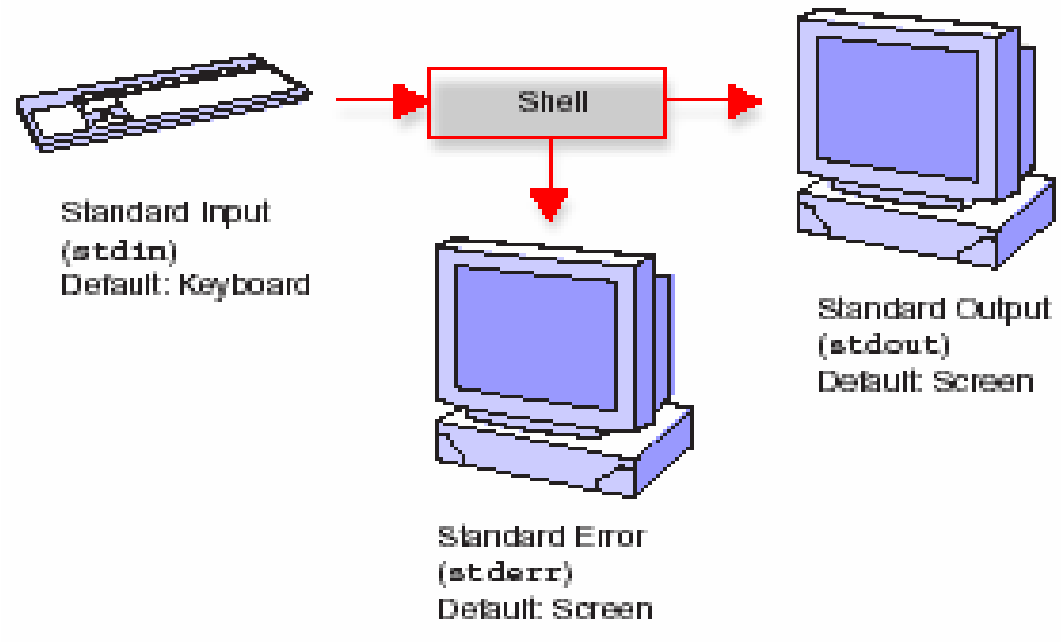


Figure 6-1 Input, CPU, and Output



Redirecting Input (stdin)

PRONET

VNU-ITP

UNDP

- Command format:

  command **<** *filename*

  command ***0<*** *filename*

- Example:

  $ **mailx user1 < ~/dante**

Redirecting Output (stdout)

PRONET
VNU-ITP
UNDP

- Command format:

  command **>** *filename*

  command **1>** *filename*

- Example:

```
$ ls -l > directory_content.list
```

Redirecting `stdout` in Append Mode

PRONET

VNU-ITP

UNDP

- Command format:

  command **>>** *filename*

- Example:

```
$ cat /etc/passwd > my_file;cat my_file
$ echo "That's my passwd file" >>
  my_file ; cat my_file
```

Redirecting stderr

- Command format:

  command **2>** *filename*

  command **2>** */dev/null*

- Example:

```
$ Date 2> errorfile
$ cat errorfile
ksh: Date: not found
```

Piping

PRONET
VNU-ITP
UNDP

- The shell enables you to pass the *output* of one command to the *input* of another command. This connection is known as a pipe (|).

- Command format:

```
command | command
```

- Examples:

```
$ head -20 yadda.txt | tail +11
$ ls -l | more
```

Background jobs

PRONET
VNU-ITP
UNDP

- Programs can also be started *"in the background"* from the command line by placing an ampersand (&) after the command:

```
$ find / -name cool_prog &

[1] 4463
```

- Programs run in background will still generate output to foreground. It's best to redirect the output to file :

```
$ find / -name cool_prog > output &
```

Managing Jobs

PRONET
VNU-ITP
UNDP

Command	Value
`jobs`	Display which jobs are currently running.
`fg %n`	Place a job in the foreground.
`bg %n`	Place a job in the background.
`kill %n`	Abort the specified background job. The job ID must be specified.
`Ctrl-c`	Abort the foreground job.
`Ctrl-z`	Suspend the foreground job.

Background jobs

PRONET

VNU-ITP

UNDP

- You can switch a foreground job to background jobs and vice versa
- Foreground to background:

  - Suspend it by `Ctrl+Z`

  - Type `bg %<job_id>`
- Background to foreground:

  - List all running jobs to get job_id, type `jobs`

  - Type `fg %<job_id>`

Bash Scripting

PRONET

VNU-ITP

UNDP

- Any number of executable commands from the command line can be placed in a text file. If this file is given *execute* permissions, it can be run like any program by type on command line. This is called a *script.*
- Bash scripts must begin with the following line :

```
#!/bin/bash
```

- Example bash script that uses conditions

```
#!/bin/bash
if [ $# -lt 1 ]; then
    echo Usage: $0 directory_name
    exit
fi
if [ -d "$1" ]; then
    echo Directory $1 already exists
else
    mkdir $1
fi
```

Variables

PRONET
VNU-ITP
UNDP

- As in most programming languages, bash allows the use of variables. Variables are used to hold temporary of program.
- There are also several global variables available that are set by the shell : #0-$9, $#, ..
- See : "*bash programming*" books

Summary

PRONET
VNU-ITP
UNDP

- Define command line basic procedures
- List common commands
- Describe and use the commands `su` and `sudo`
- Describe and perform basic file system tasks
- Describe the Linux file system, its structure, and directory hierarchy

Summary

PRONET

VNU-ITP

UNDP

- Locate commands and files
- Identify Linux text editors (e.g *vi*)
- Describe and use the Linux shell environment (e.g `bash`)
- Describe and use text processing tools and filters