

THE UNIX OPERATING SYSTEM

SE 101

Spiros Mancoridis

WHAT IS AN OS?

- An operating system (OS) is software that manages the resources of a computer
- Like most managers, the OS aims to manage its resources in a safe and efficient way
- Examples of computer resources are: CPU, RAM, disk memory, printers, displays, keyboard, mouse, etc
- The OS also isolates users and application programmers from the underlying computer

OPERATING SYSTEMS

Microsoft Windows



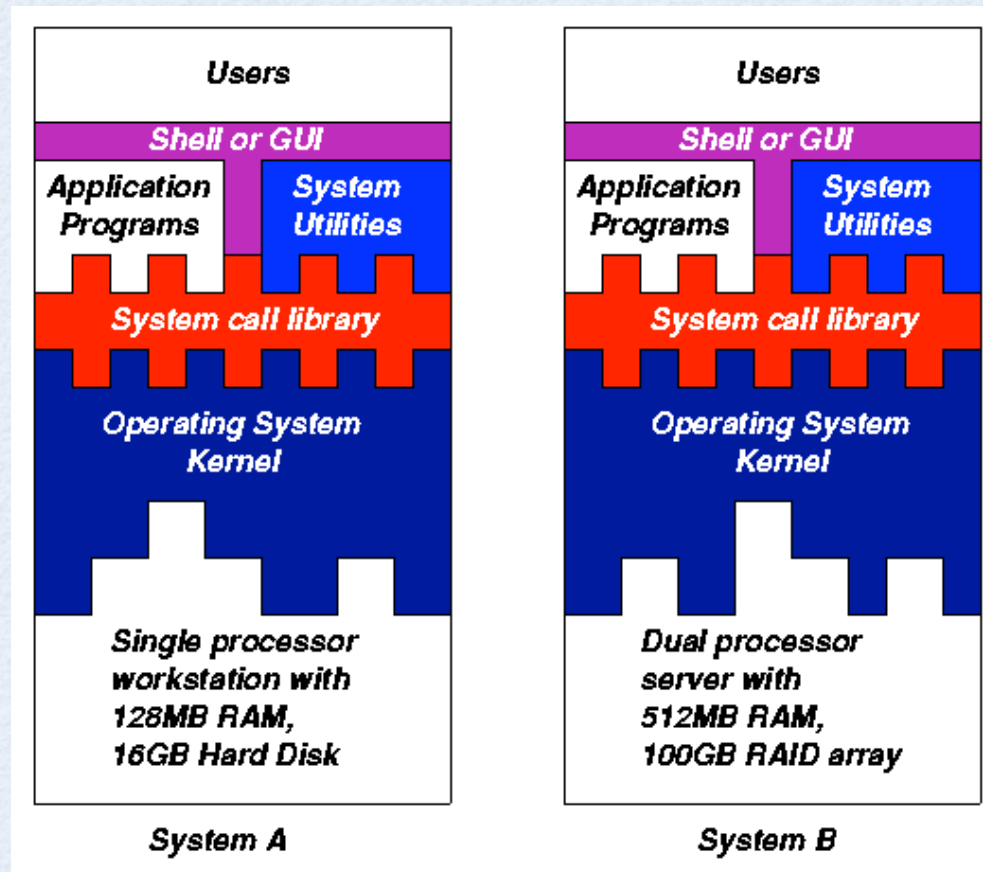
search ID: aba0612

"When I said you will get an office with windows, I was referring to the operating system on the computer."

Unix



OS ARCHITECTURE



Without an OS, every application would have to implement some part of this software hierarchy ...

UNIX

- A popular multi-user, multi-tasking OS
- Attributes: stability, portability, security
- Created at Bell Labs by Dennis Ritchie and Ken Thompson (won the ACM Turing Award in 1983)
 - Unix is considered one of the greatest achievements in computer science
- Has been around since the 1960s in various forms, e.g., AIX, SCO Unix, SunOS, FreeBSD, OpenBSD, NetBSD, Linux, Mac OS X

UNIX

- Multiuser and Multitasking
- Toolbox philosophy
- Concise syntax
- Designed by programmers for programmers

1983 ACM TURNING AWARD (UNIX)

- ACM is the Association for Computing Machinery
 - World's largest educational and scientific computer society
 - You can become a student member too
www.acm.org
- The ACM awards the Turing Award every year. It is the “Nobel Prize” of computing
- Named after british mathematician Alan M. Turing (1912-1954)



Thompson and Ritchie



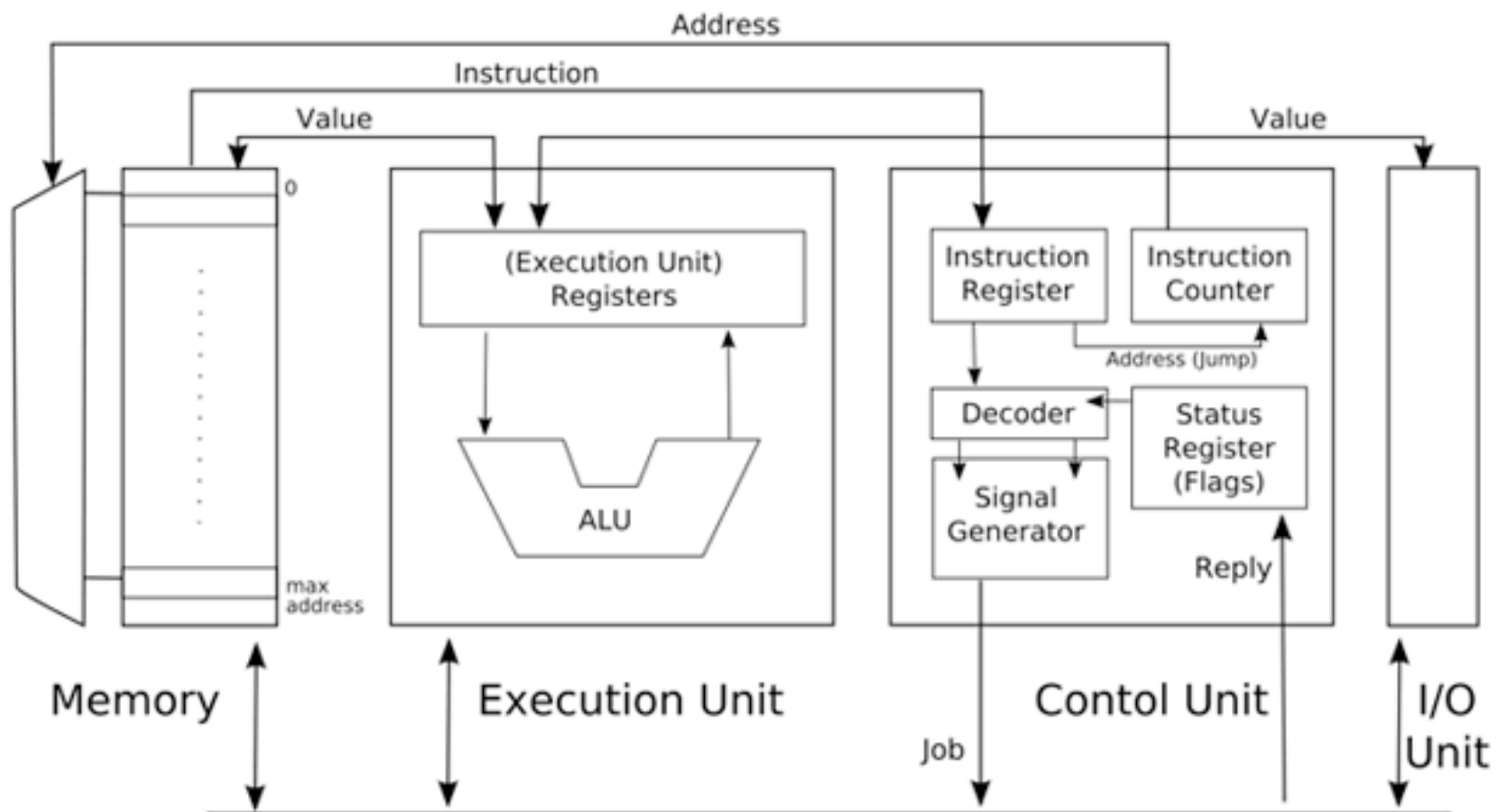
Alan M. Turing

UNIX KERNEL

- Includes device drivers for computer hardware devices, e.g., graphics cards, network cards, disks
 - A device driver is a program that allows computer programs to interact with hardware devices
- CPU and memory management
- File system management
- Implements system calls that can be used by application programs and system utilities

WHAT'S INSIDE THE BOX?

- The von Neumann Architecture
- The OS hides this complexity from the programmer



UNIX SHELLS AND GUIs

- Shells are used for command line input/output to and from users
 - e.g., **sh** (Bourne shell), **bash** (Bourne again shell), **csh** (C shell), **ksh** (Korn shell)
- GUIs are used for graphical I/O
 - e.g., Linux KDE, GNOME, Mac OS Leopard

```
bash-2.05b$ pwd
/home/dstone
bash-2.05b$ cd /usr/portage/app-shells/bash
bash-2.05b$ ls -al
total 60
drwxr-xr-x  3 root root 4096 May 14 12:05 .
drwxr-xr-x 26 root root 4096 May 12 02:36 ..
-rw-r--r--  1 root root 13710 May  3 22:25 ChangeLog
-rw-r--r--  1 root root 2324 May 14 12:05 Manifest
-rw-r--r--  1 root root 3720 May 14 12:05 bash-2.05b-r11.ebuild
-rw-r--r--  1 root root 3516 May  2 20:05 bash-2.05b-r9.ebuild
-rw-r--r--  1 root root 5003 May  3 22:25 bash-3.0-r11.ebuild
-rw-r--r--  1 root root 4030 May 14 12:05 bash-3.0-r7.ebuild
-rw-r--r--  1 root root 3731 May 14 12:05 bash-3.0-r0.ebuild
-rw-r--r--  1 root root 4267 Mar 29 21:11 bash-3.0-r9.ebuild
drwxr-xr-x  2 root root 4096 May  3 22:25 files
-rw-r--r--  1 root root 164 Dec 29 2003 netdata.xml
bash-2.05b$ cat netdata.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pkgmetadata SYSTEM "http://www.gentoo.org/dtd/netdata.dtd">
<pkgmetadata>
  <herd>base-system</herd>
</pkgmetadata>
bash-2.05b$ sudo /etc/init.d/bluetooth status
Password:
* status: stopped
bash-2.05b$ ping -q -c1 en.wikipedia.org
PING rr.chtpa.wikimedia.org (207.142.131.247) 56(84) bytes of data:

--- rr.chtpa.wikimedia.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 112.076/112.076/112.076/0.000 ms
bash-2.05b$ grep -l /dev/sda /etc/fstab | cut -f1 -d=
/dev/sda1
/dev/sda2
bash-2.05b$ date
Wed May 25 11:36:56 PDT 2005
bash-2.05b$ lsmod
Module                  Size  Used by
jbd2                    8256  0
ipu2200                 175112  0
ieee80211               44228  1 ipu2200
ieee80211_crypt         4072  2 ipu2200, ieee80211
e1000                   84468  0
bash-2.05b$
```



UNIX SYSTEM UTILITIES

- System utilities are specialized software tools (commands) e.g.,
 - `ls`, `cp`, `grep`, `awk`, `bc`, `wc`, `more`, `rm`, `mkdir`, ...
- Daemons provide remote network and administration services e.g.,
 - `ssh` (remote login)
 - `lpd` (remote printing)
 - `httpd` (serves web pages)

UNIX APPLICATION PROGRAMS

- This is the software that users commonly interact with e.g.,
 - **vi** and **emacs** (text editors)
 - **gcc** (GNU C compiler)
 - **javac** (Java compiler)
 - **java** (Java run time virtual machine system)

WHAT IS A PROGRAMMING LANGUAGE?

- A programming language is an artificial language designed to express computations that will be executed on a computer
- Programming languages have a syntax (form) and semantics (meaning)
- Java and C are example programming languages

WHAT IS A COMPILER?

A compiler is a computer program that transforms human readable source code of another computer program into machine readable code that a computer can execute.

The act of transforming source code into machine code is called *compilation*.

```
package com.javadb.examples;

import java.util.Arrays;
import java.util.List;
import java.util.Iterator;

public class Main {

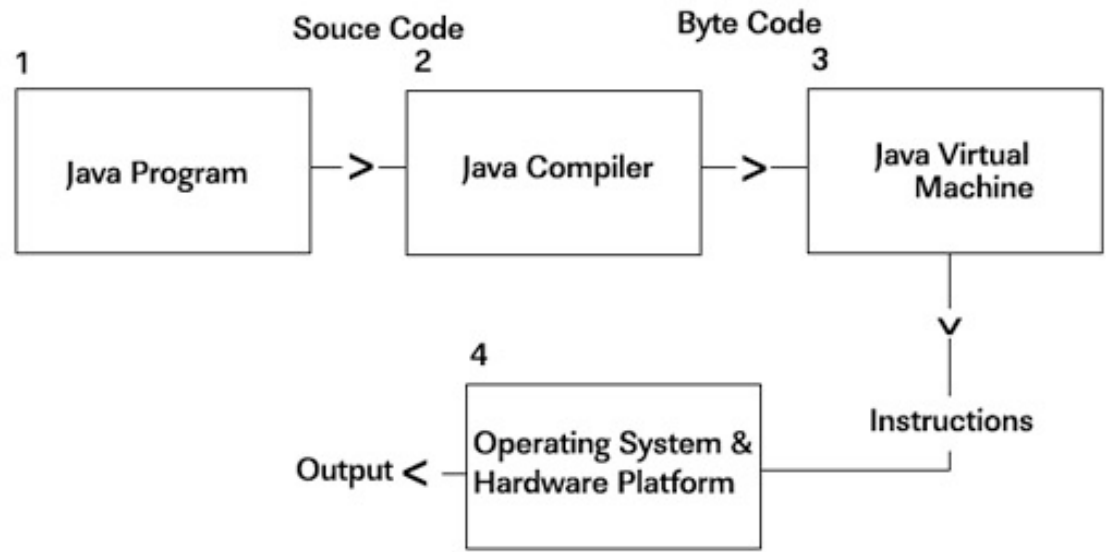
    public static void main(String[] args) {

        String[] array = {"Programming", "is", "cool"};
        List<String> list = Arrays.asList(array);

        Iterator<String> iterator = list.iterator();
        while (iterator.hasNext()) {
            System.out.println(iterator.next());
        }
    }
}
```

Java Source Code

Java Program Execution



Java Compilation Process

WHAT IS A VIRTUAL MACHINE?

- A virtual machine (VM) is a software implementation of a computer that executes programs like a physical computer
- A system VM implements a complete computer that can support the execution of a real OS (e.g., vmware, KVM)
- A process VM is designed to run a single program (e.g., Java VM)
 - Improves program portability, i.e., the ability to reuse software on a different platform with little or no modification

LOGGING INTO UNIX

login: <type your user id>

password: <type your password>

\$ pwd

/home/spiros

\$ exit

LOGGING INTO UNIX

- userid is the unique name of the user
- password can be changed
- password should only be known by user
- system staff does not know user password, but they have the power to change it
- Unix is case sensitive

PASSWORDS ... DO

- make sure no one is looking while you enter your password
- change your password often
- choose a password that you can remember
- use at least 8 characters (letters) in your password
- Mix letters (upper and lower case) with digits and punctuation

PASSWORDS ... DON'T

- use a word that can be found in a dictionary
- use a name
- keep your password written on a piece of paper
- use control characters
- use personal information (e.g., address, birthday, name of pet)
- ever give your password to anybody

CHANGING PASSWORD

- Use the Unix **passwd** command
- You will be prompted for both your current password (once) and your new password (twice)
- change your initial password immediately

LOGGING OUT OF UNIX

- `logout` leaves the system
- `exit` quits the shell
- `^D` same as `logout`, but often disabled

UNIX COMMANDS

- **\$ command** *-options targets*
- **man man** (a manual on the Unix manual)
 - There should be a man page for every Unix command e.g., **man ls**
- Read the man pages, they are very useful and convenient

SOME UNIX COMMANDS

passwd

date

hostname

who

last

finger

w

clear

cal

bc -l

history

ssh

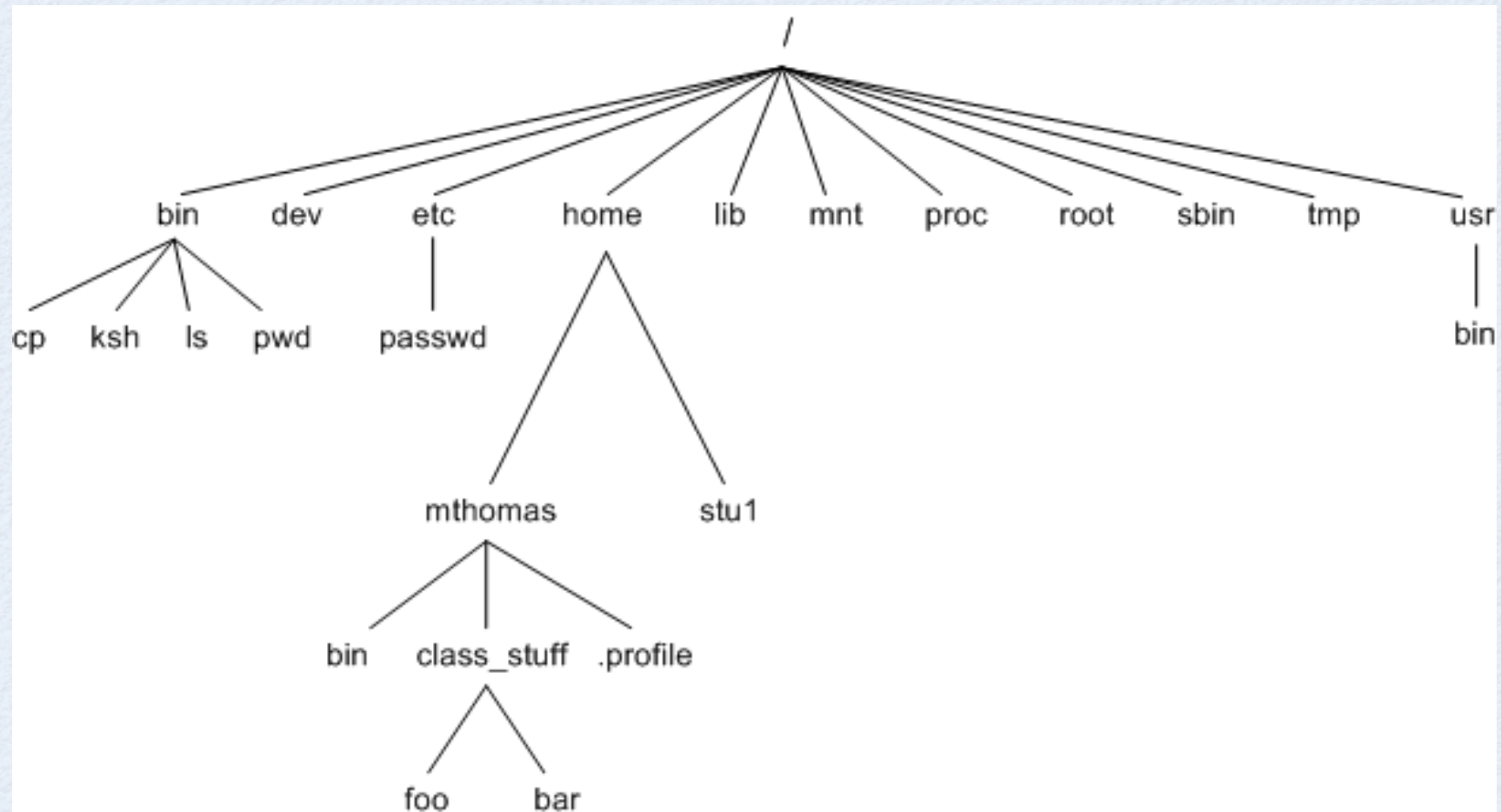
Try these Unix
commands and see
what they do ...

The man pages can
give you details on
how to use these (and
other) commands

UNIX FILESYSTEM

- Ordinary Files
 - Files contain data, program code, etc
 - File names cannot have the '/' character in them
- Directories
 - Contain files and other directories
- Links
 - A link is a pointer reference to another file (like an alias)
- Devices
 - Allows applications to access hardware devices

UNIX FILESYSTEM HIERARCHY



UNIX DIRECTORIES

/	The "root" directory
/bin	Essential low-level system utilities
/usr/bin	Higher-level system utilities and application programs
/sbin	Superuser system utilities (for performing system administration tasks)
/lib	Program libraries (collections of system calls that can be included in programs by a compiler)for low-level system utilities
/usr/lib	Program libraries for higher-level user programs
/tmp	Temporary file storage space (can be used by any user)
/home	User home directories containing personal file space for each user. Each directory is named after the login of the user.
/etc	UNIX system configuration and information files
/dev	Hardware devices
/proc	A pseudo-filesystem which is used as an interface to the kernel. Includes a sub-directory for each active program (or process).

USEFUL UNIX FILESYSTEM COMMANDS

\$ **cd** /usr/bin

\$ **pwd**
/usr/bin

\$ **cd** /

\$ **ls**

bin/ lib/ media/ proc/ selinux/ sys/ var/ boot/ dev/ homes/ lib32/ mnt/
root/ site/ tmp/ etc/ opt/ sbin/ srv/ usr/

\$ **man** ls

LS -L

- An example of the output is of `ls -l` is:

```
drwxr-xr-x 3 spiros serg 238 May 5 2:05 license.dat
```

- *type* is a single character which is either 'd' (directory), '-' (ordinary file), 'l' (symbolic link).
- *permissions* is a set of characters describing access rights. There are 9 permission characters, describing 3 access types given to 3 user categories. The three access types are read ('r'), write ('w') and execute ('x'), and the three users categories are the user who owns the file, users in the group that the file belongs to and other users (the general public). An 'r', 'w' or 'x' character means the corresponding permission is present; a '-' means it is absent.
- *links* refers to the number of filesystem links pointing to the file or directory.
- *owner* is the user who created the file or directory.
- *group* denotes a collection of users who are allowed to access the file according to the group access rights specified in the permissions field.
- *size* is the length of a file, or the number of bytes used by the operating system to store the list of files in a directory.
- *date* is the date when the file or directory was last modified. The `-u` option display the time when the file was last accessed (read).
- *name* is the name of the file or directory.

USEFUL UNIX FILESYSTEM COMMANDS

- **cd** *path*
(change directory to *path*)
- **mkdir** *directory*
(make a new directory)
- **rmdir** *directory*
(remove a directory)
- **cp** *source-file destination-file*
(copy source-file into destination-file)
- **cp** *source-file(s) destination-directory*
(copy source files into destination-directory)

USEFUL UNIX FILESYSTEM COMMANDS

- **mv** *source destination*
(move / rename source file or directory to destination file or directory)
- **rm** *file(s)*
(remove / delete files)
- **rm -rf** *directory*
(remove entire directory)
- **cat** *target-file(s)*
(concatenate target files and display them on the screen)
- **cat** *target-file(s)* > output.txt
(store concatenation to output file)

USEFUL UNIX FILESYSTEM COMMANDS

- **ln -s *filename linkname***
(create a pointer to filename and call it linkname)
- **cat ?piros**
(concatenates all files that start with any character and end with piros)
- **cat ***
(concatenates all files in the current directory)
- **ls [a-c]*[x-z]**
(lists files that start with a letter from a-c and end with a letter from x-z)

FILE AND DIRECTORY PERMISSIONS

- **chmod** *options files*
 - for options u (user), g (group), o (other), a (all), r (read), w (write), x (execute), + (add permission), - (remove permission), = (assign permission)
- What does **chmod** *ug=rw, o-rw, a-x *.txt* do?
- How about **chmod -R go+r dir** ?
- How about **chmod 600 private.txt** ?
- **chgrp** *group file(s)*
(changes the group permissions for the files, works with -R option)

UNIX FILE COMMANDS

- **file** *filename(s)*
(reports on the type of a file e.g., text, HTML, Java source code)
- **head** *-num filename*
(displays the first *num* lines of a file)
- **tail** *-num filename*
(displays the last *num* lines of a file)
- **tail** *-f filename*
(continuously outputs the last few lines of a file being modified)

THE UNIX FIND COMMAND

- **find** *directory* -name *targetfile* -print
(look for *targetfile* in any part of the directory rooted at *directory*), e.g.,
 - **find** /home -name "*.java" -print
- **which** *command*
(gives path where the executable code of *command* resides), e.g.,
 - **which** ls (should return /bin/ls)

UNIX GREP COMMAND

- **grep** *options pattern files*
(searches the files for lines that match a given pattern). e.g.,
- **grep** Drexel *.txt
- **grep** Drexel `find . -name "*.txt" -print`
- **grep -i** drexel *.txt

UNIX SORT COMMAND

- **sort** *filename(s)*
(sorts line in a group of concatenated files alphabetically)
- **sort -n** *filename(s)*
(sorts line in a group of concatenated files numerically)
- **uniq** *filename*
(removes duplicate adjacent lines from a file)
- **sort** in1.txt in2.txt > out.txt

UNIX FILE ARCHIVING

- **tar -cvf** *archivename filenames*
(create archive from a list of files or directories)
- **tar -tvm** *archivename*
(list contents of an archive)
- **tar -xvf** *archivename*
(extract and restore archive)
- **tar -cvf** code.tar src
- **tar -xvf** code.tar

UNIX FILE COMPRESSION

- **zip** *filename.zip directory / **
(compresses contents of directory and store them in *filename.zip*)
- **unzip** *filename.zip*
(uncompress contents of *filename.zip*)
- Other Unix compression tools are:
 - **gzip**
 - **compress**

UNIX PROCESSES

- You can get the status and process id of each process running on the system
- Check the man pages because options for **ps** may vary from one Unix system to another
- **ps auxw**
- **ps -ef**
- To terminate a process use the **kill** command
 - **kill -9 *process-id***

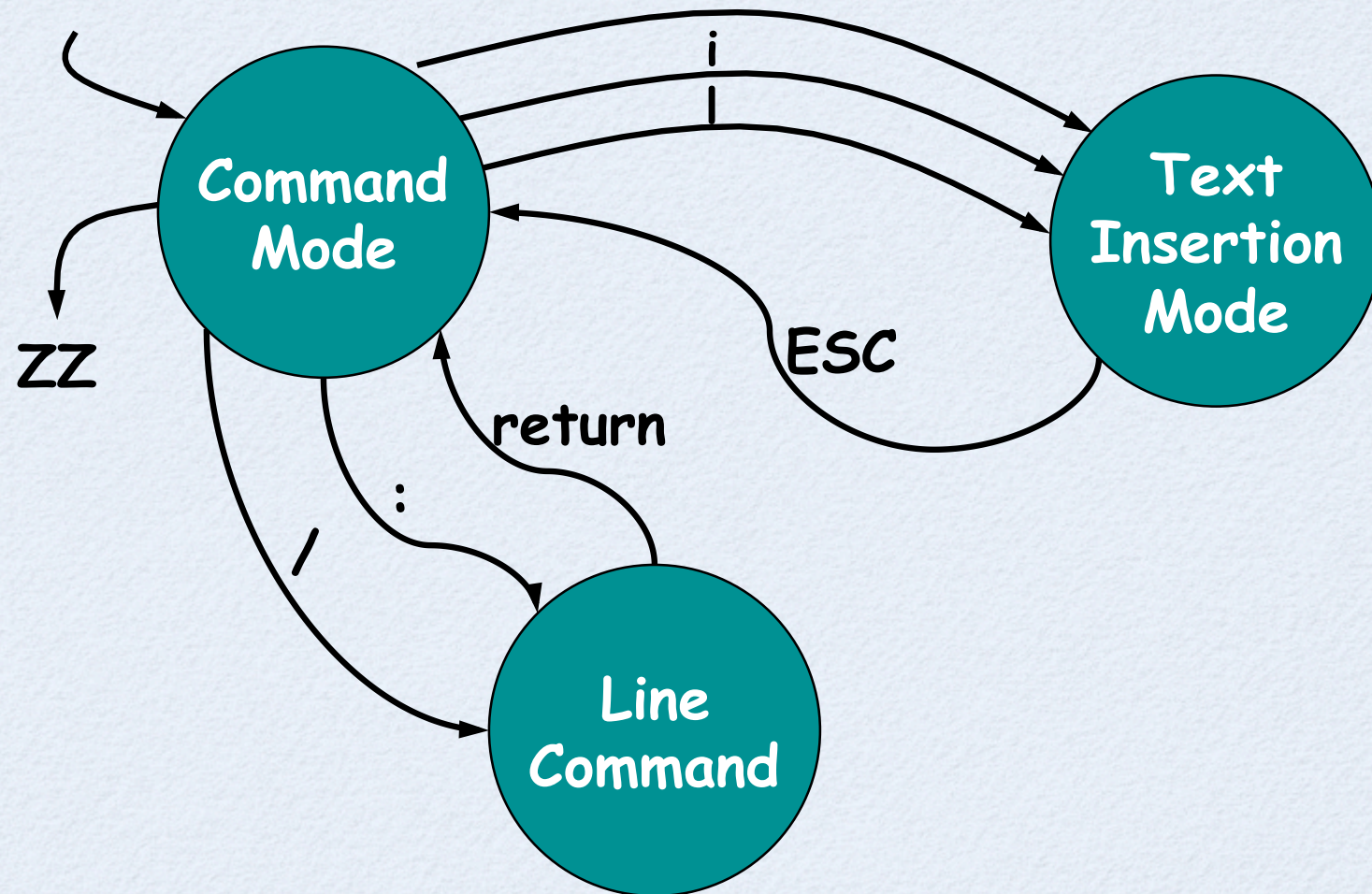
PRINTING

- **lpr** -P*printerid* *filename*
- **lpq** -P*printerid* [*job#*] [*userid*]
- **lprm** -P*printerid* [*job#*] [*userid*]

THE VI TEXT EDITOR (PRONOUNCED V-I)

- To run **vi** type the following on the command line:
 - `$ vi filename`
- **vi** has three modes:
 - **command mode:**
 - to navigate through the document
 - **insert mode:**
 - to add text to the document
 - **command line mode:**
 - to perform manipulations on the files (e.g., search, save)

THE THREE MODES OF VI



MOVING THE CURSOR

- You can use your arrow keys
- Or you can use the h, j, k, l keys
 - h move left one space
 - j move down one space
 - k move up one space
 - l move right one space
- You can move faster by typing 5 h to move left 5 spaces, etc

MOVING THE CURSOR

- ^F go forward in the document by 1 screen
- ^B go backward in the document by 1 screen
- ^D go down in the document by half a screen
- ^U go up in the document by half a screen
- Note that ^ means hold the control key down e.g., ^F means hold the control key down and press F. The F,B,D,U characters are non case sensitive

MOVING THE CURSOR

- **G** go to the last line in the file
- ***n* G** go to the n^{th} line in the file
- **\$** go to the end of the current line
- **^** go to the beginning of current line
(use carat key not control key)
- **0** same as **^**, go to beginning of current line
- **w** forward one word, use ***n* w** to go forward ***n*** words
- **b** backward one word, use ***n* b** to go backward ***n*** words
- **e** go to end of the word

INSERTING TEXT

- **i** insert text before the cursor
- **a** append text after the cursor
- **I** insert text at the beginning of line
- **A** append text at the end of line
- **o** open new line after current line
- **O** open new line before current line

DELETING TEXT

- **dd** delete current line
- ***n* dd** delete *n* lines starting from the current line
- **dw** delete word
- ***n* dw** delete *n* next words
- **D** delete from cursor to the end of current line
- **x** delete current character
- ***n* x** delete next *n* characters
- **X** delete previous character (backspace)

CHANGING TEXT

- **cw** change current word
- ***n* cw** change *n* words starting at current word
- **c\$** change from cursor to the end of line
- **~** change case of character
- **J** join next line to the current line
- **u** undo last command
- ***n* u** undo last *n* commands

CHANGING TEXT

- `.` repeat last change
- `yy` yank current line into a storage buffer (copy)
- `n yy` yank next `n` lines into a storage buffer
- `yw` yank current word into a storage buffer
- `n yw` yank next `n` lines into a storage buffer
- `p` put yanked buffer text (or deleted text) after cursor
- `P` put yanked buffer text (or deleted text) before cursor

MANIPULATING FILES

- `:w` write file to disk (save)
- `:wq` write file to disk (save), then exit vi (quit)
- `:w!` force overwrite of file
- `:q` quit vi if no changes have been made
- `:q!` quit vi without saving any of the changes
- `:! command` escape to shell and run command
- `:r! command` insert the result of command at current cursor position