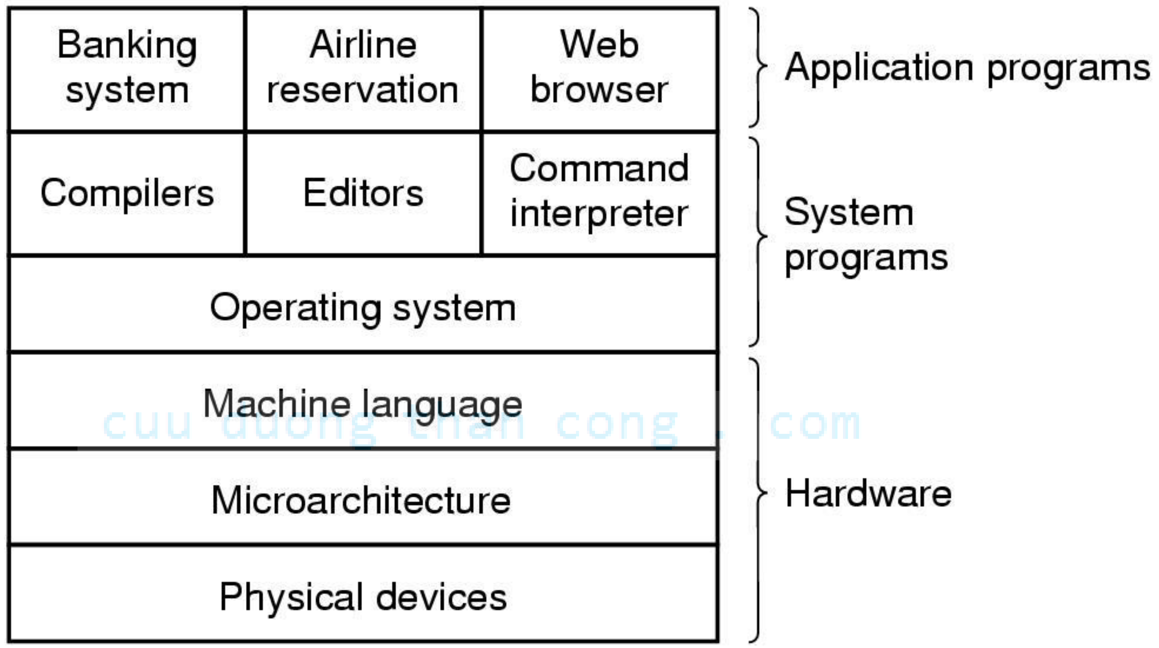
# Chapter 1

# Introduction

1.1 What is an operating system
1.2 History of operating systems
1.3 The operating system zoo
1.4 Computer hardware review
1.5 Operating system concepts
1.6 System calls
1.7 Operating system structure

# Introduction

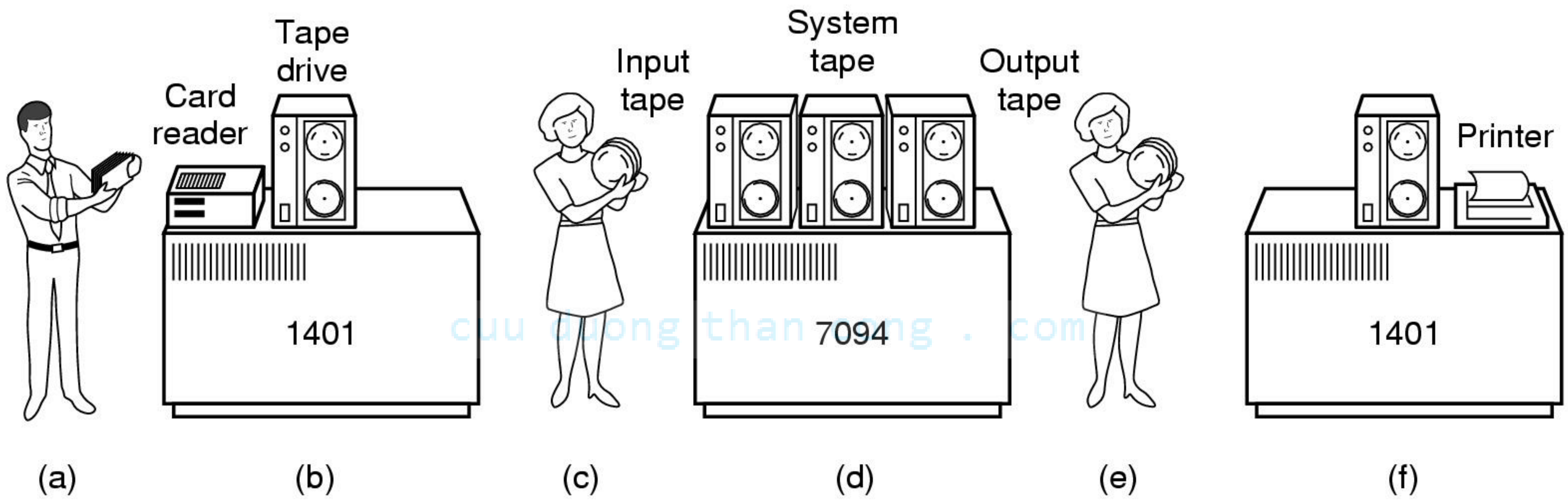


- A computer system consists of
  - hardware
  - system programs
  - application programs

## What is an Operating System

- It is an extended machine
  - Hides the messy details which must be performed
  - Presents user with a virtual machine, easier to use
- It is a resource manager
  - Each program gets time with the resource
  - Each program gets space on the resource

# History of Operating Systems (1)



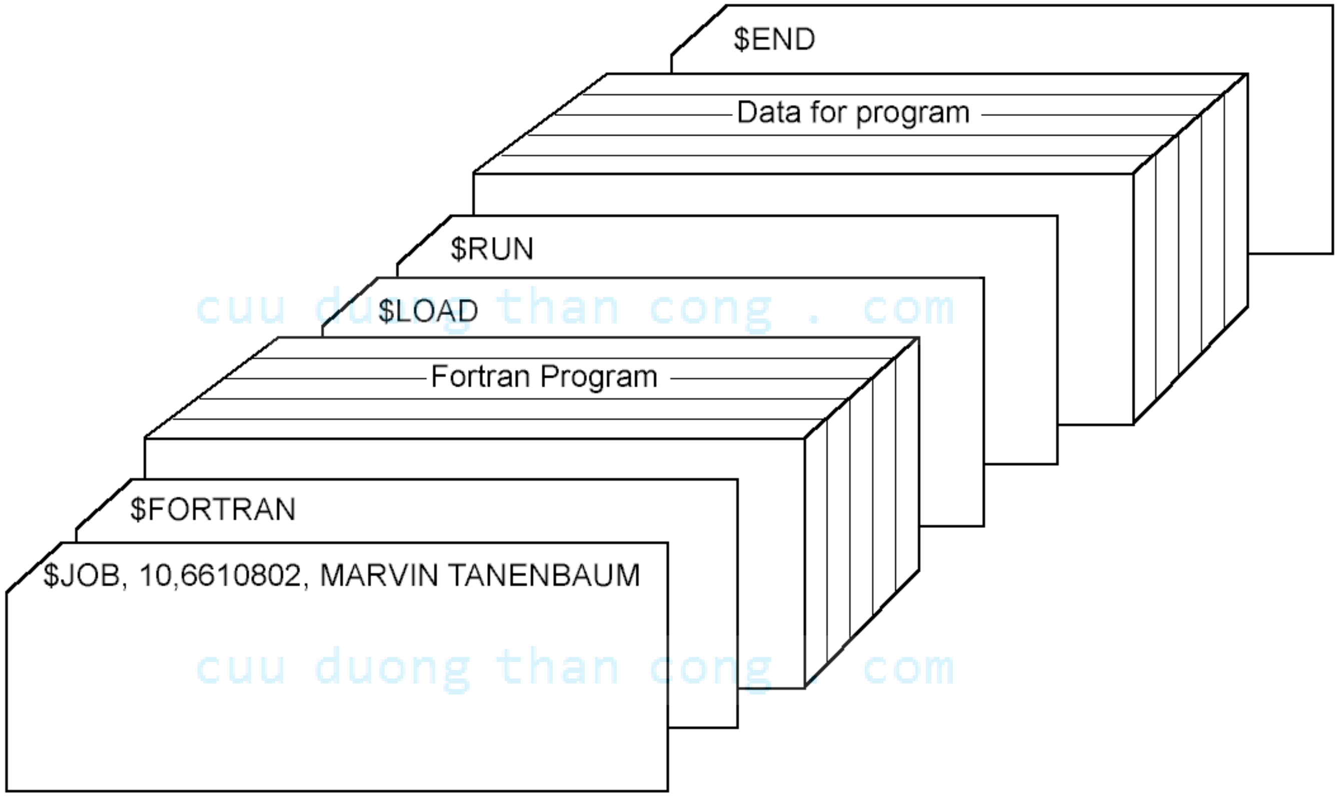
Early batch system

- bring cards to 1401
- read cards to tape
- put tape on 7094 which does computing
- put tape on 1401 which prints output

## History of Operating Systems (2)

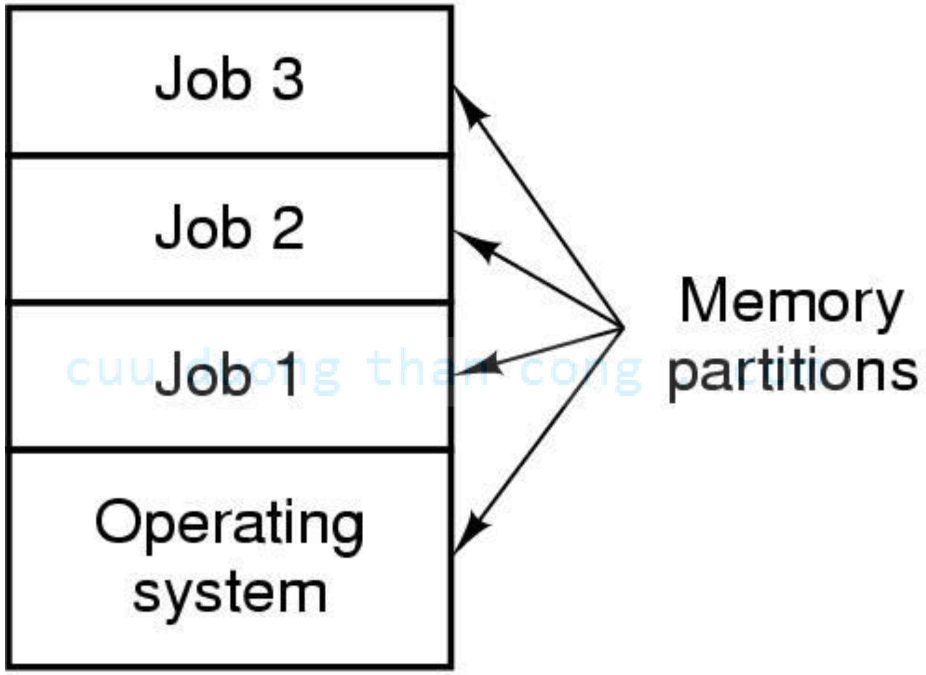
- First generation 1945 - 1955
  - vacuum tubes, plug boards
- Second generation 1955 - 1965
  - transistors, batch systems
- Third generation  1965 – 1980
  - ICs and multiprogramming
- Fourth generation 1980 – present
  - personal computers

# History of Operating Systems (3)



- Structure of a typical FMS job – 2nd generation

# History of Operating Systems (4)

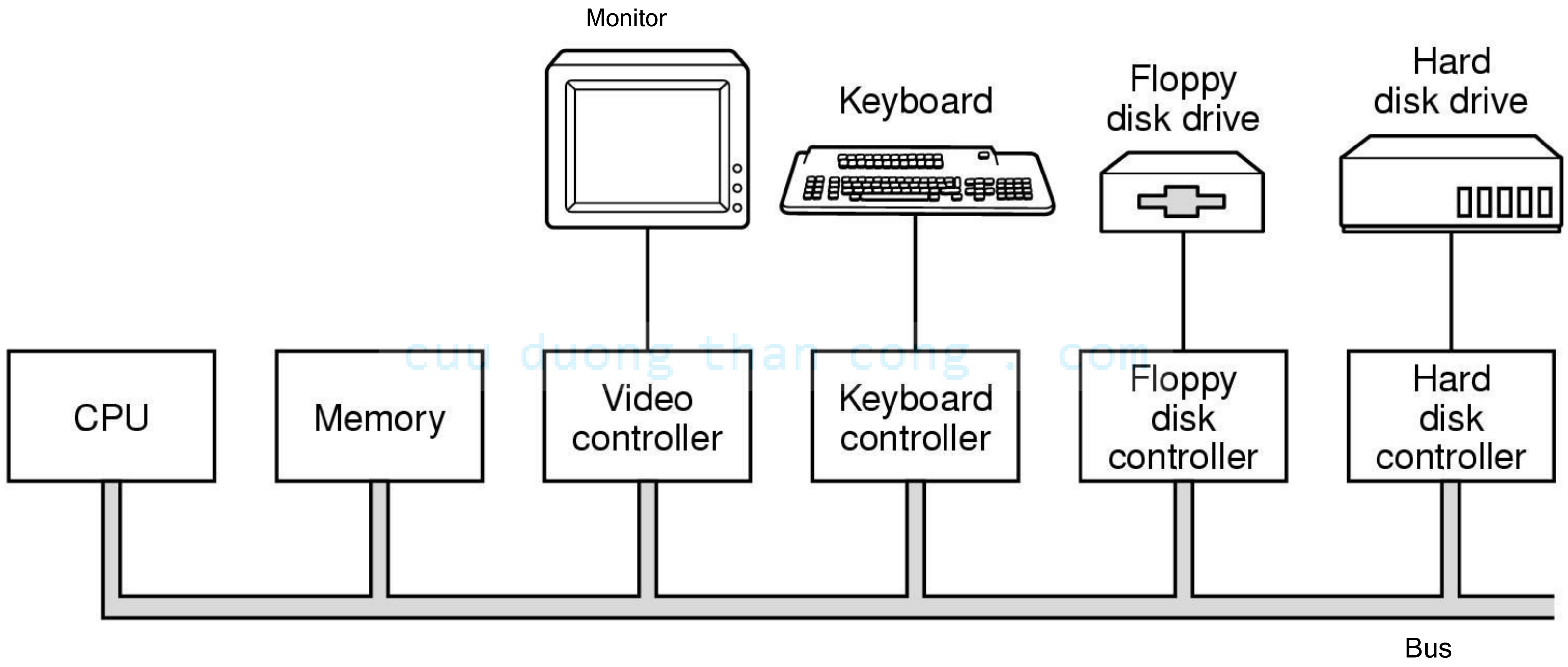


- Multiprogramming system
  - three jobs in memory – 3rd generation

# The Operating System Zoo

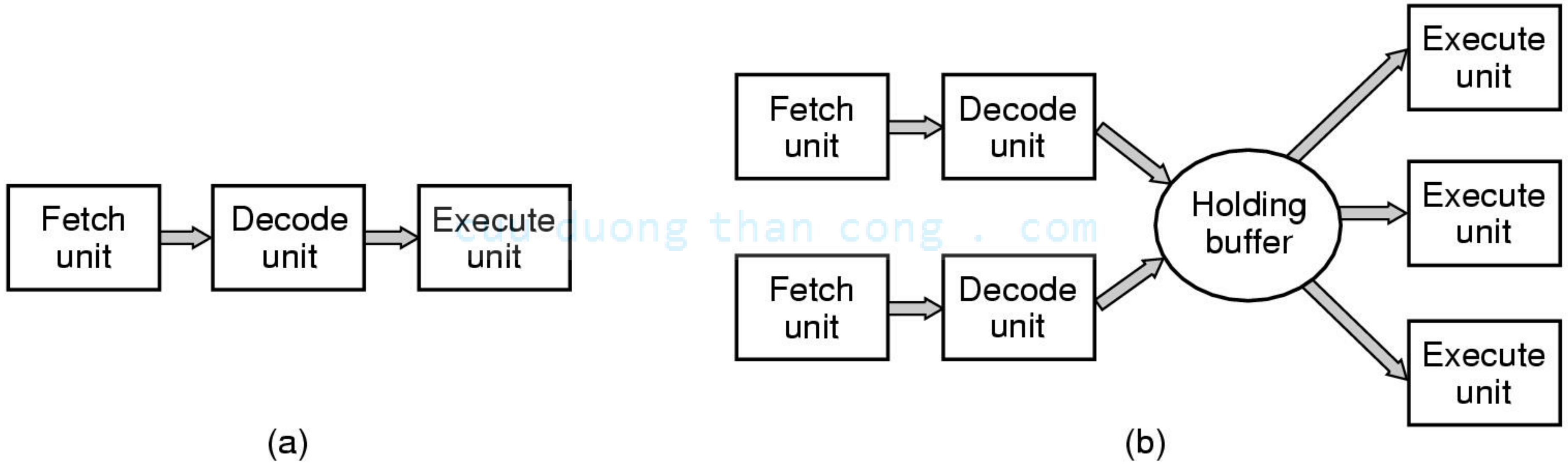
- Mainframe operating systems
- Server operating systems
- Multiprocessor operating systems
- Personal computer operating systems
- Real-time operating systems
- Embedded operating systems
- Smart card operating systems

# Computer Hardware Review (1)



- Components of a simple personal computer

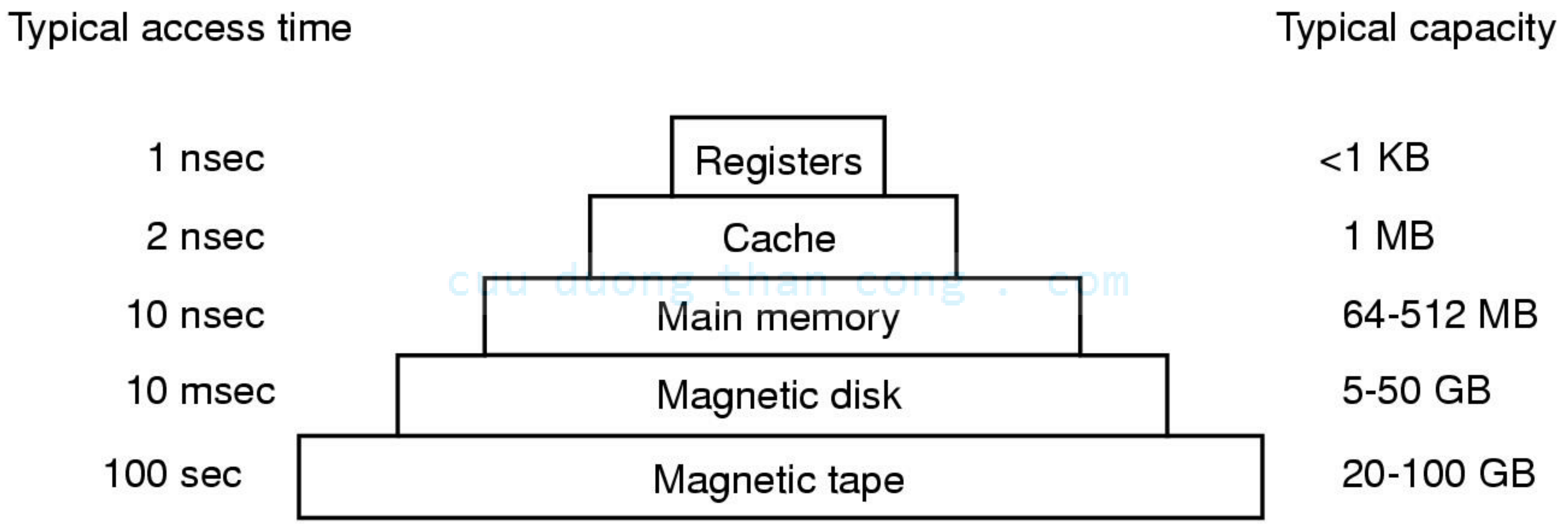
# Computer Hardware Review (2)



(a) A three-stage pipeline

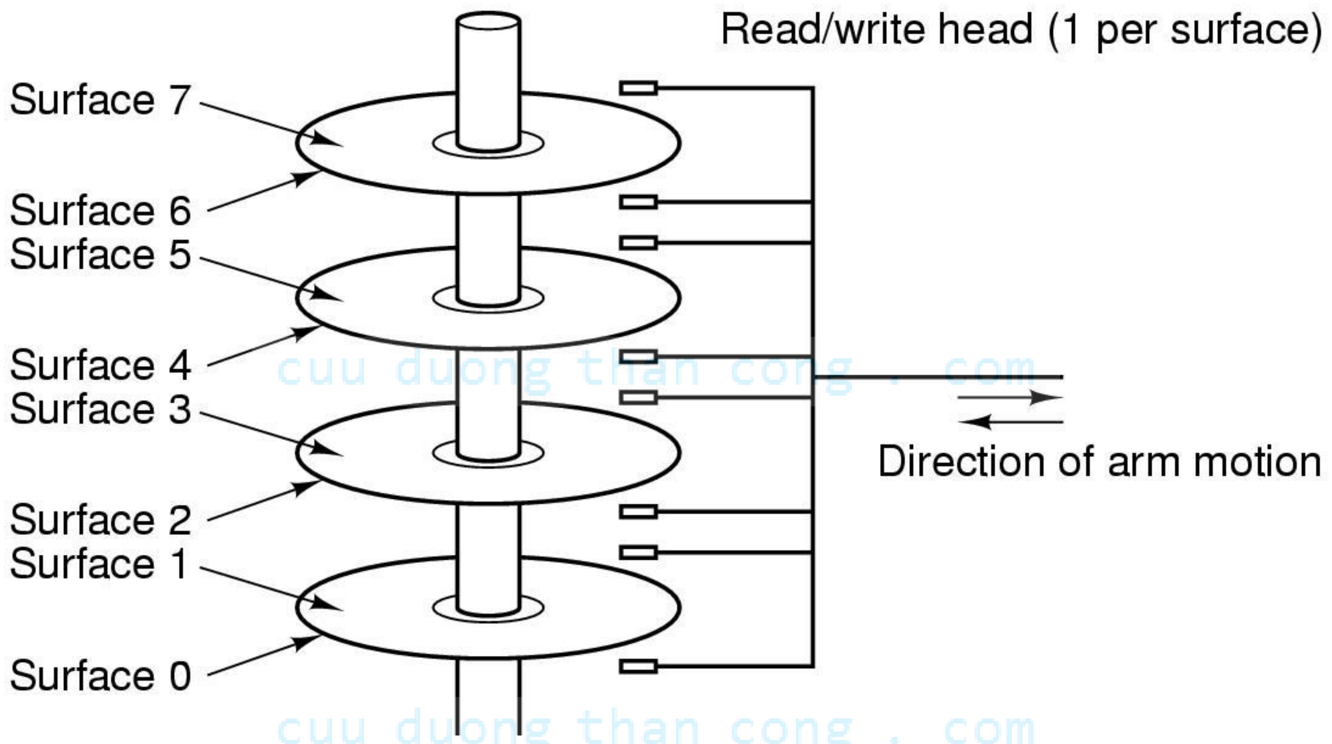
(b) A superscalar CPU

# Computer Hardware Review (3)



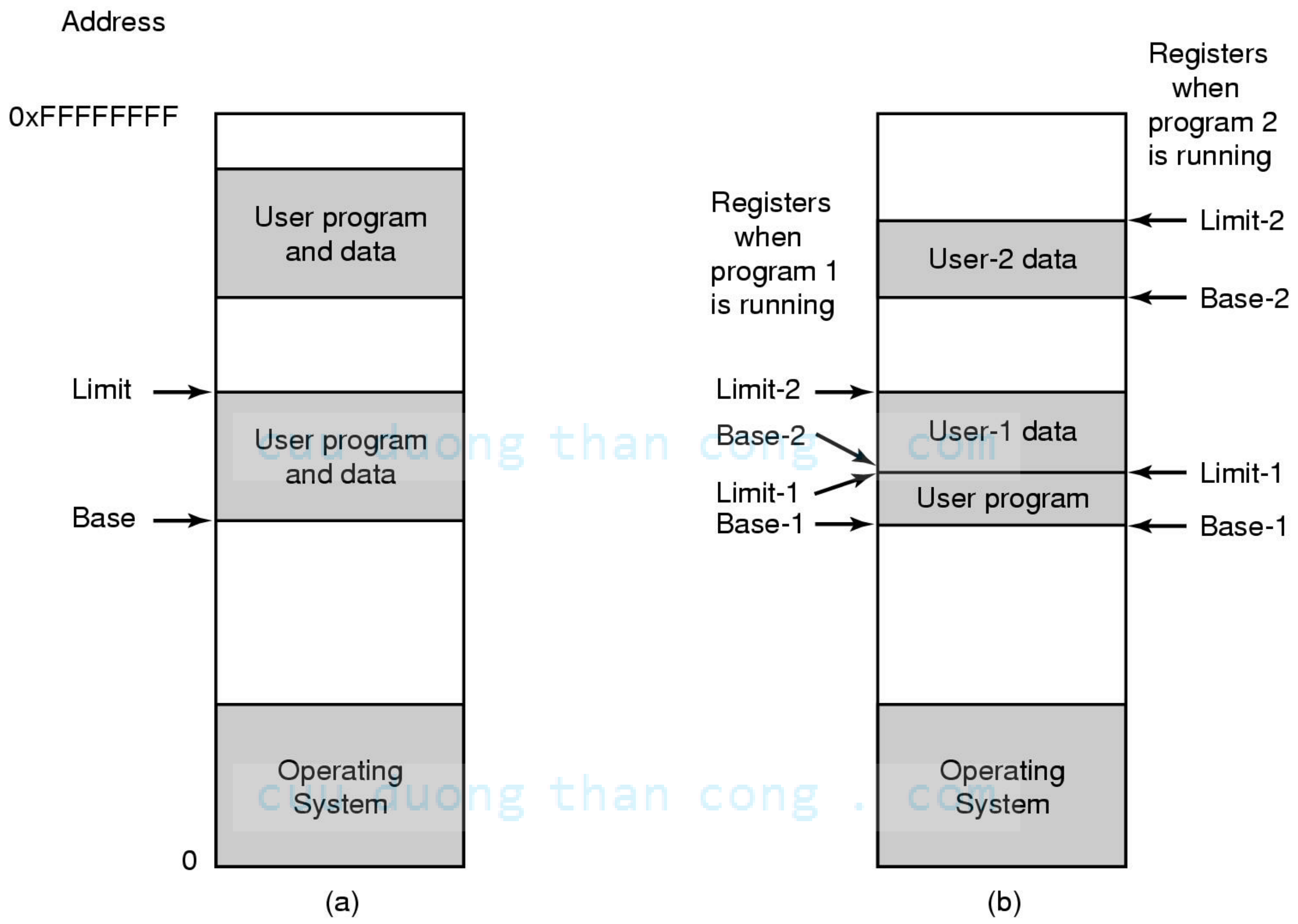
- Typical memory hierarchy
  - numbers shown are rough approximations

# Computer Hardware Review (4)



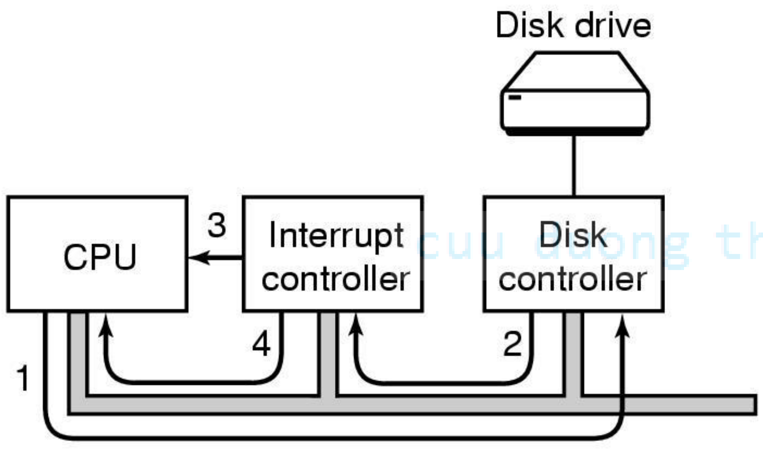
Structure of a disk drive

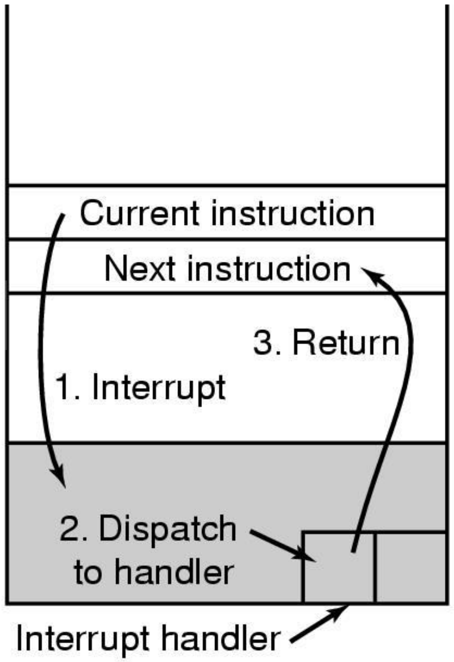
# Computer Hardware Review (5)



One base-limit pair and two base-limit pairs

# Computer Hardware Review (6)



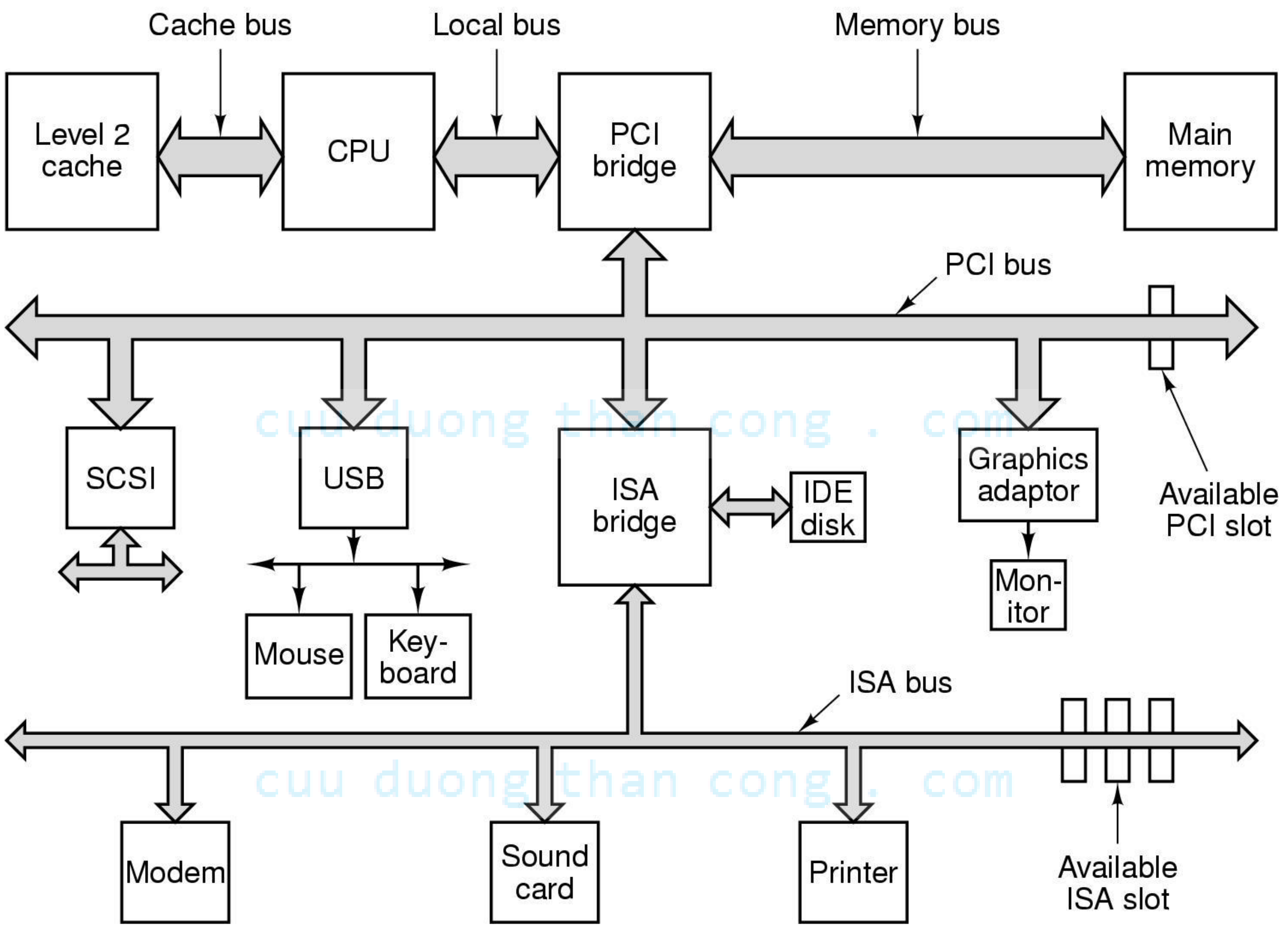


(a) (b)

(a) Steps in starting an I/O device and getting interrupt

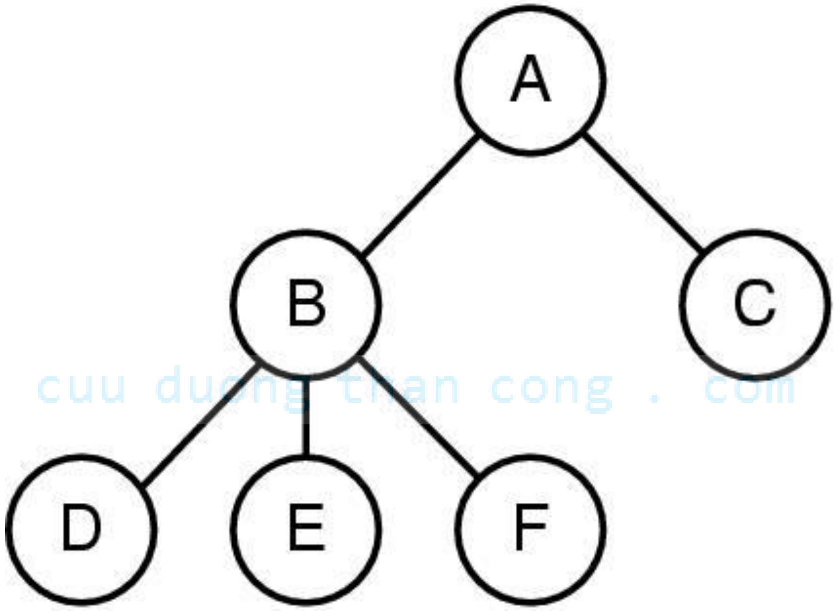
(b) How the CPU is interrupted

# Computer Hardware Review (7)



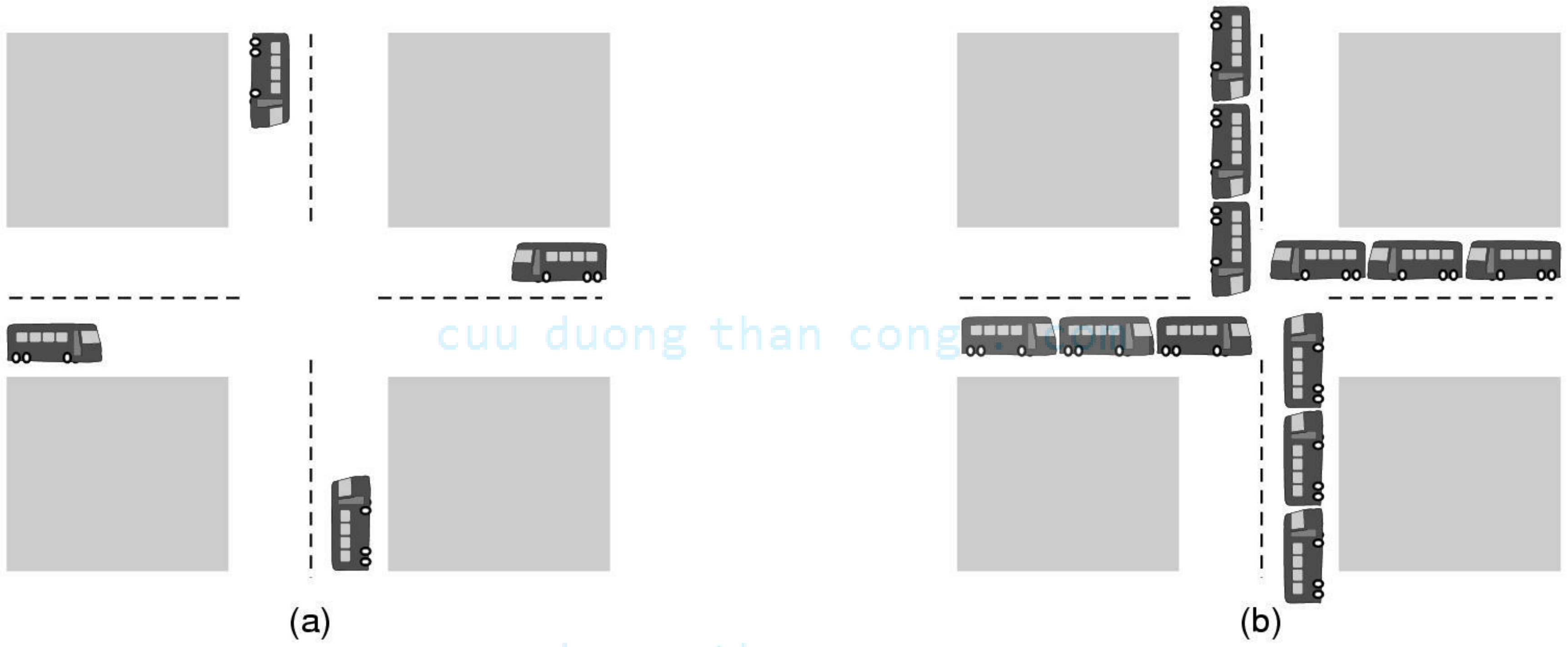
Structure of a large Pentium system

# Operating System Concepts (1)



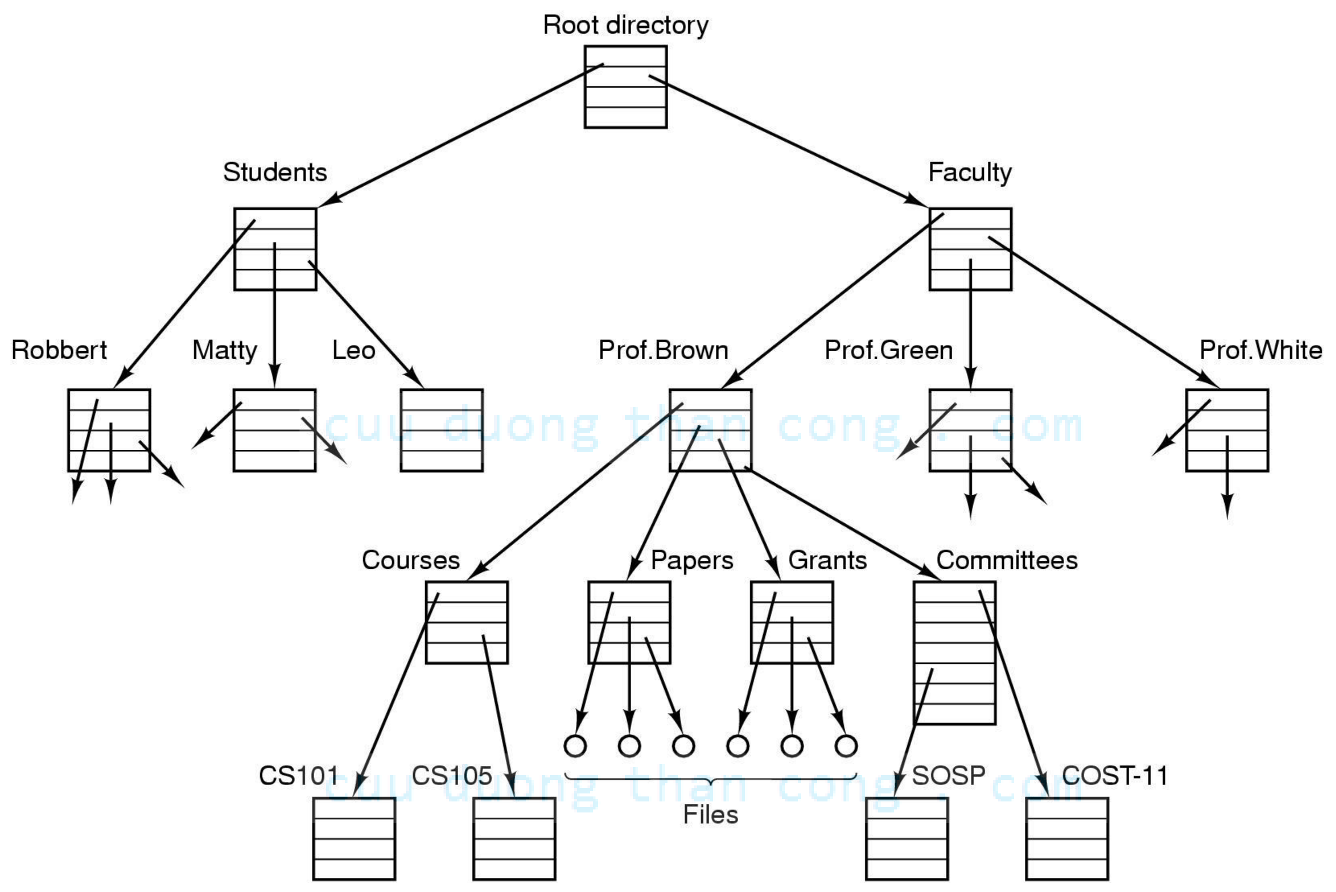
- A process tree
  - A created two child processes, B and C
  - B created three child processes, D, E, and F

# Operating System Concepts (2)



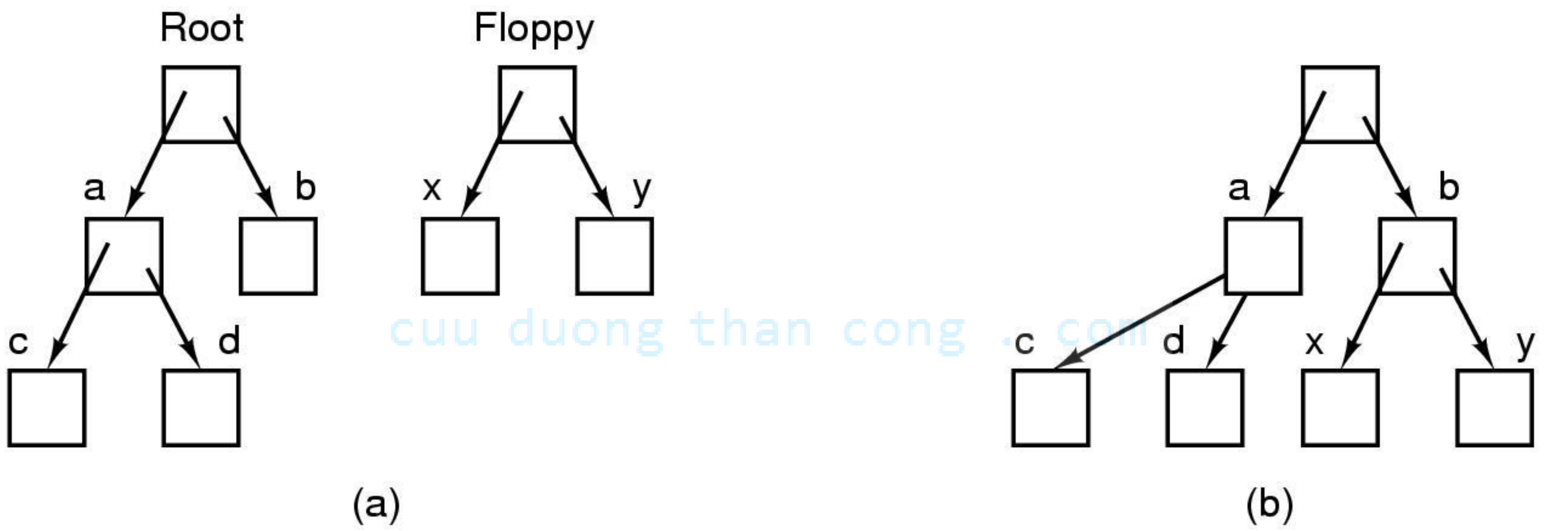
(a) A potential deadlock. (b) an actual deadlock.

# Operating System Concepts (3)



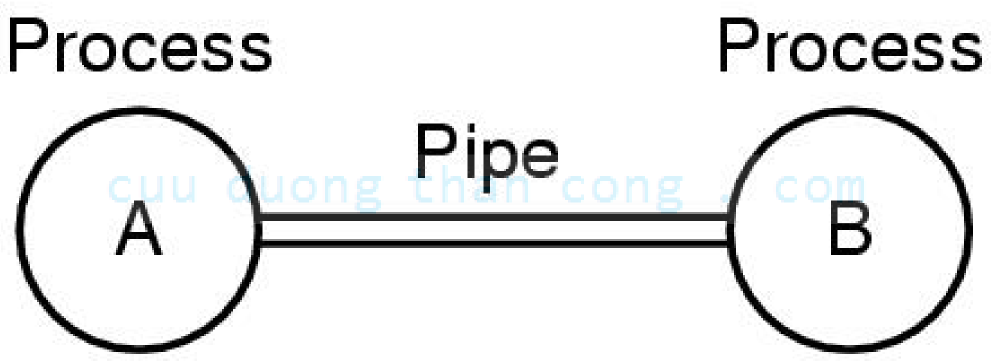
File system for a university department

# Operating System Concepts (4)



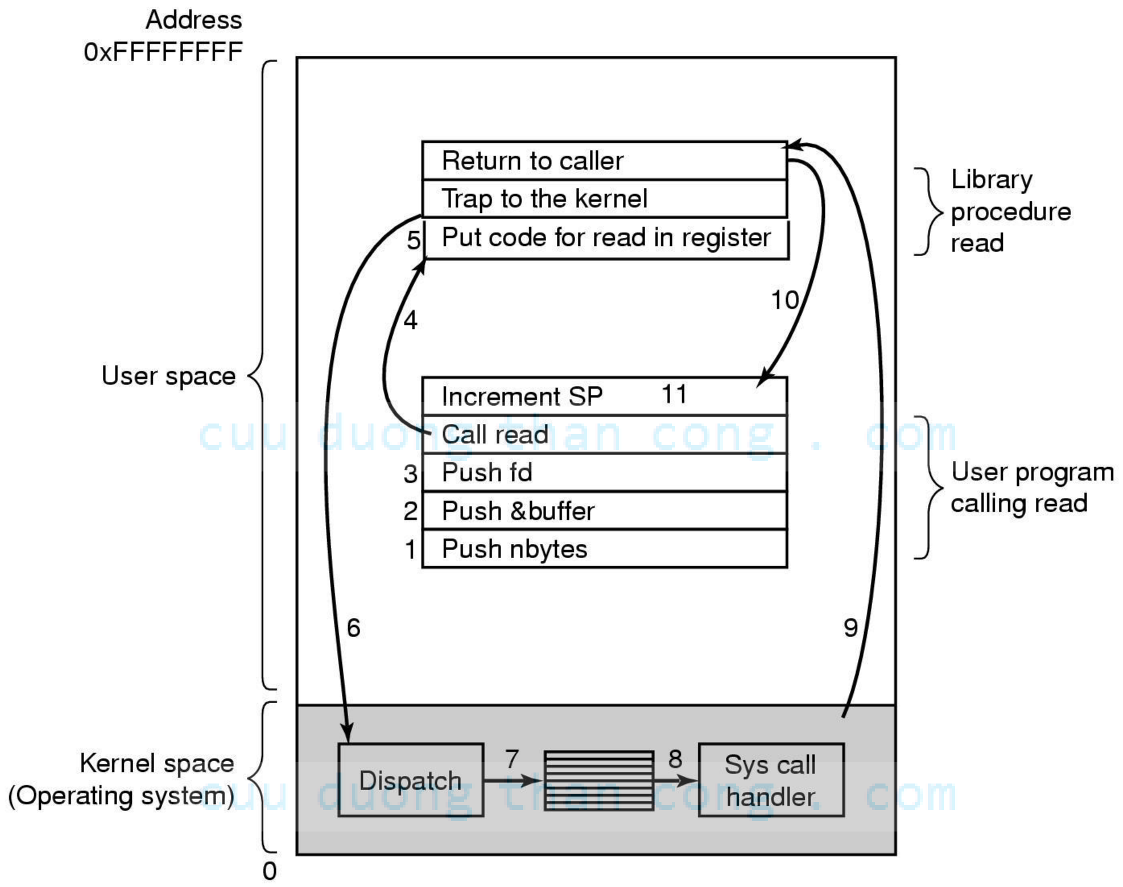
- Before mounting,
  - files on floppy are inaccessible
- After mounting floppy on b,
  - files on floppy are part of file hierarchy

# Operating System Concepts (5)



Two processes connected by a pipe

# Steps in Making a System Call



There are 11 steps in making the system call

read (fd, buffer, nbytes)

## Some System Calls For Process Management

**Process management**

| Call | Description |
|---|---|
| pid = fork( ) | Create a child process identical to the parent |
| pid = waitpid(pid, &statloc, options) | Wait for a child to terminate |
| s = execve(name, argv, environp) | Replace a process' core image |
| exit(status) | Terminate process execution and return status |

# Some System Calls For File Management

**File management**

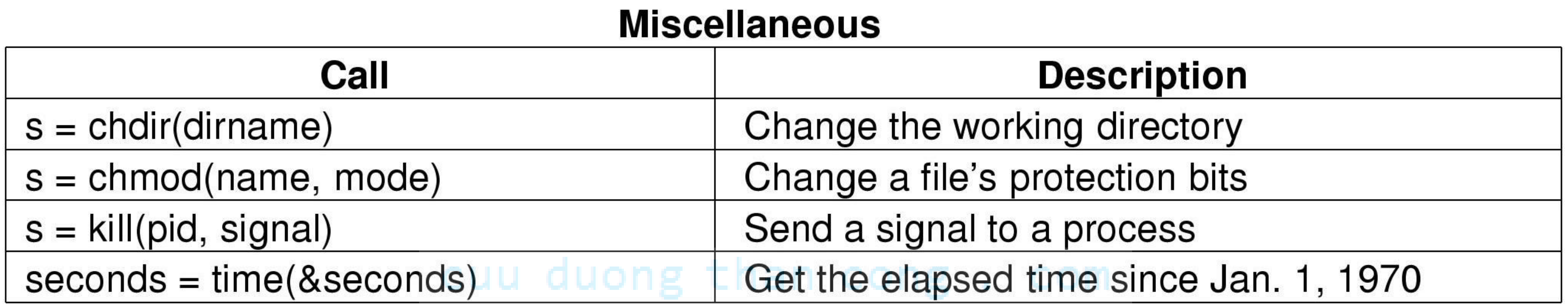
| Call | Description |
|---|---|
| fd = open(file, how, ...) | Open a file for reading, writing or both |
| s = close(fd) | Close an open file |
| n = read(fd, buffer, nbytes) | Read data from a file into a buffer |
| n = write(fd, buffer, nbytes) | Write data from a buffer into a file |
| position = lseek(fd, offset, whence) | Move the file pointer |
| s = stat(name, &buf) | Get a file's status information |

# Some System Calls For Directory Management

**Directory and file system management**

| Call | Description |
|---|---|
| s = mkdir(name, mode) | Create a new directory |
| s = rmdir(name) | Remove an empty directory |
| s = link(name1, name2) | Create a new entry, name2, pointing to name1 |
| s = unlink(name) | Remove a directory entry |
| s = mount(special, name, flag) | Mount a file system |
| s = umount(special) | Unmount a file system |

# Some System Calls For Miscellaneous Tasks



**Miscellaneous**

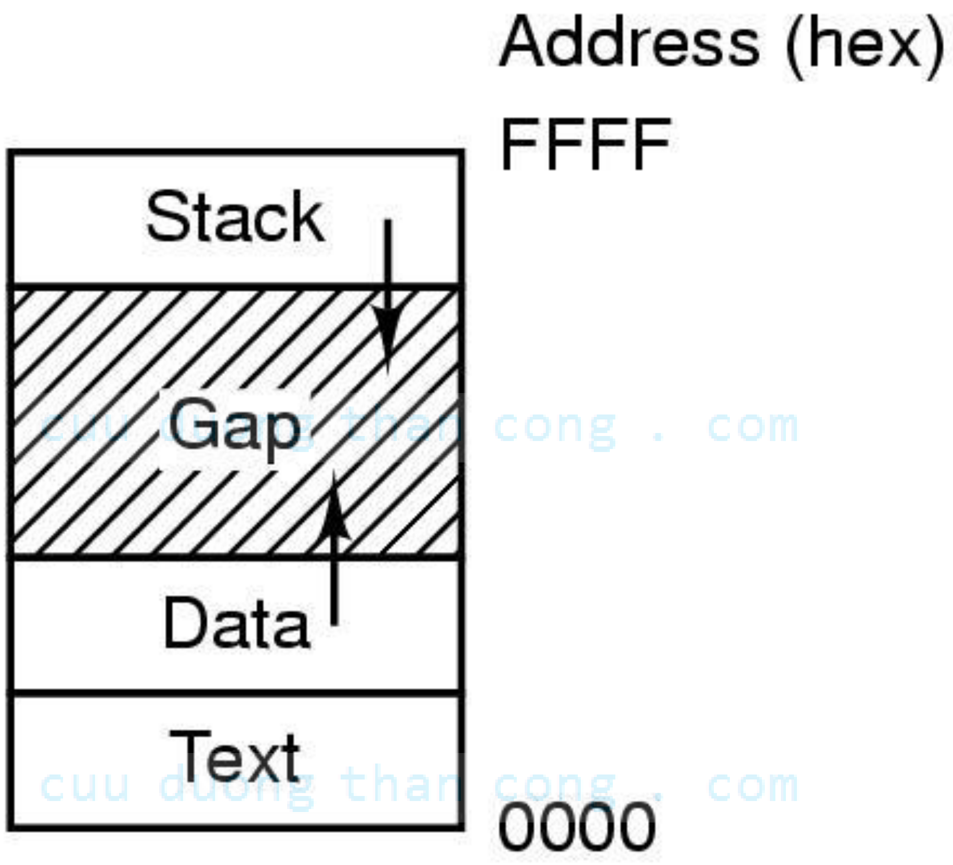
| Call | Description |
|---|---|
| s = chdir(dirname) | Change the working directory |
| s = chmod(name, mode) | Change a file’s protection bits |
| s = kill(pid, signal) | Send a signal to a process |
| seconds = time(&seconds) | Get the elapsed time since Jan. 1, 1970 |

# System Calls (1)

- A stripped down shell:

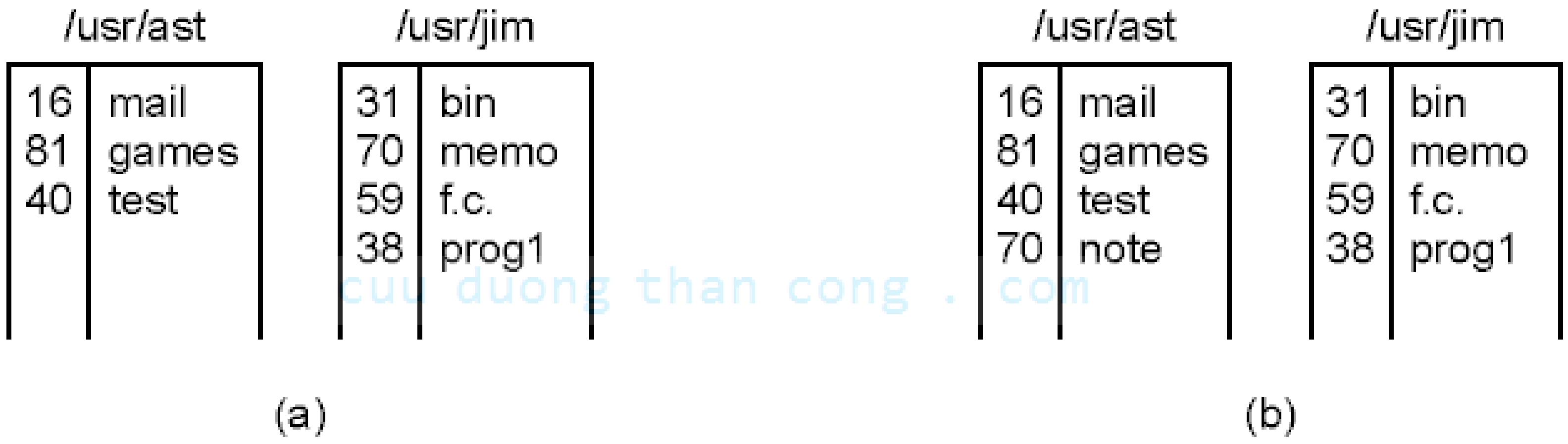
```
while (TRUE) {                                  /* repeat forever */
    type_prompt( );                             /* display prompt */
    read_command (command, parameters)          /* input from terminal */

if (fork() != 0) {                              /* fork off child process */
    /* Parent code */
    waitpid( -1, &status, 0);                   /* wait for child to exit */
} else {
    /* Child code */
    execve (command, parameters, 0);            /* execute command */
 }
}
```

## System Calls (2)



- Processes have three segments: text, data, stack

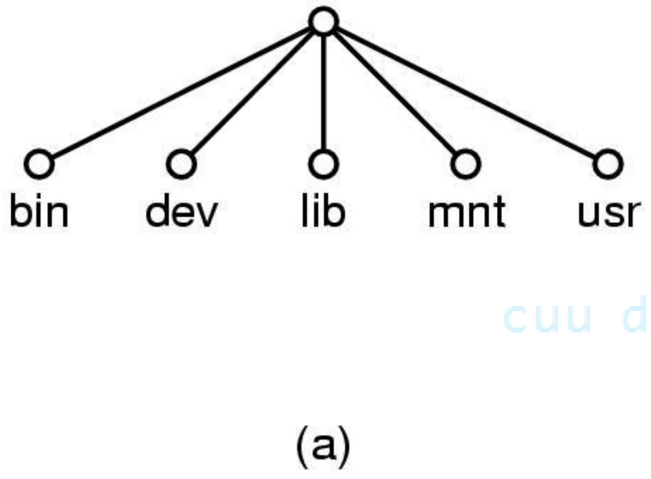
# System Calls (3)

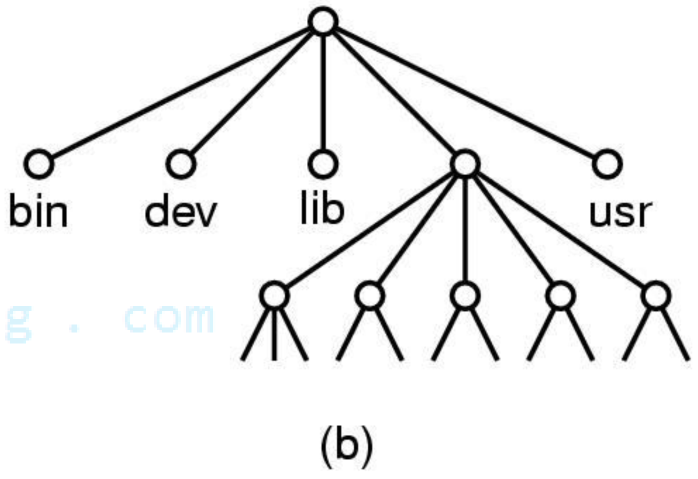


(a) Two directories before linking *~~/usr/jim/memo~~* to ast's directory

(b) The same directories after linking

# System Calls (4)

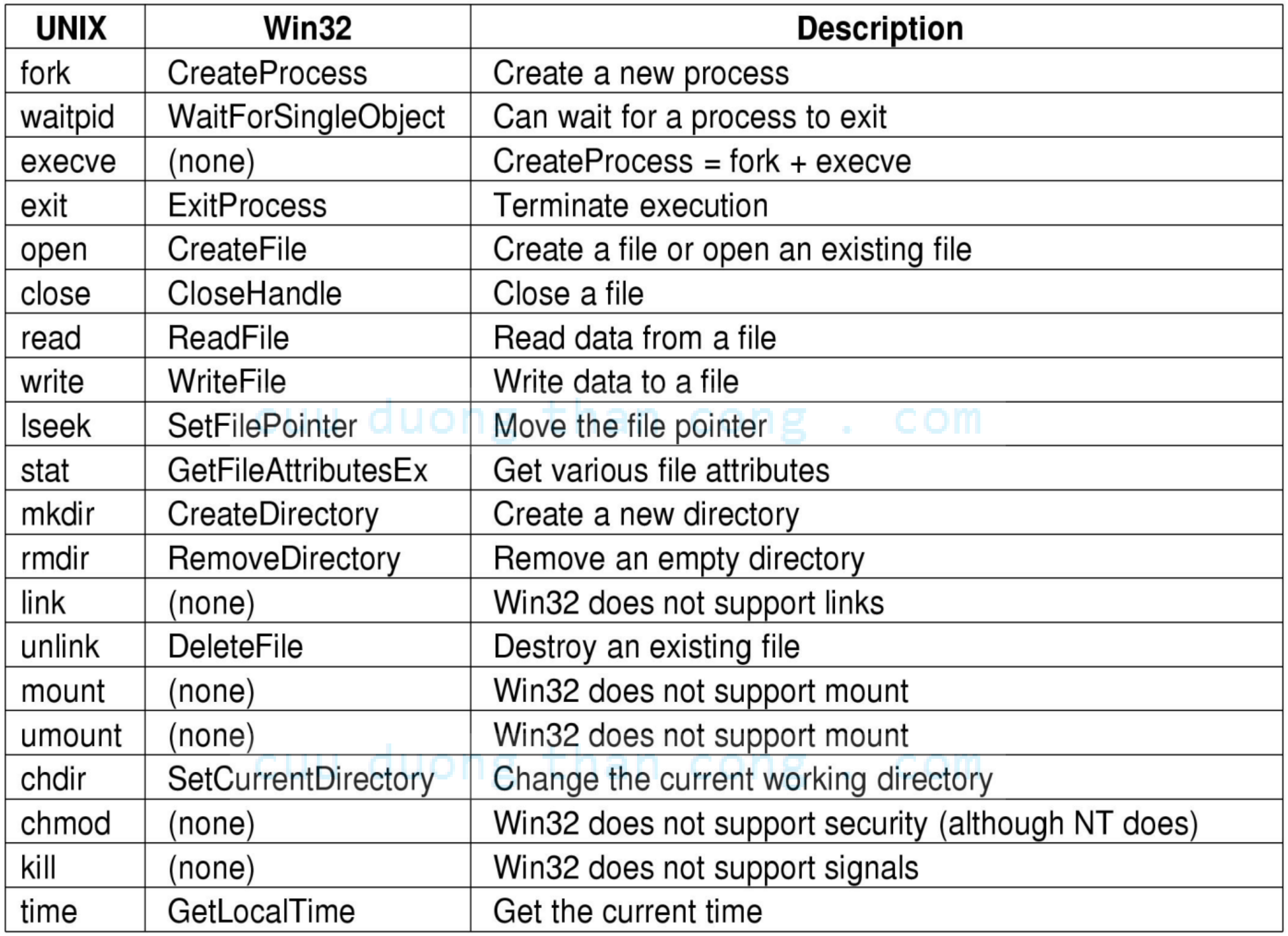




(a) File system before the mount

(b) File system after the mount

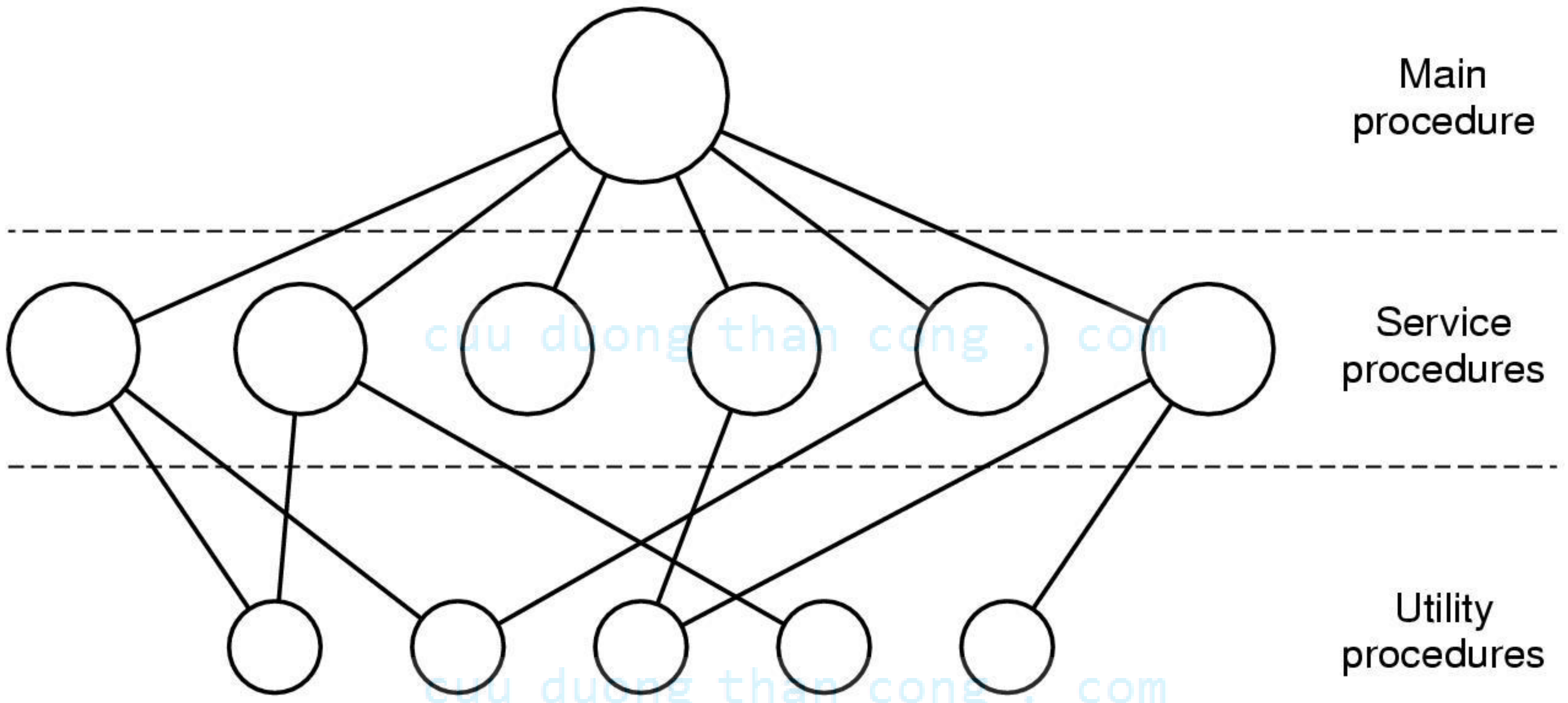
# System Calls (5)



| UNIX | Win32 | Description |
|---|---|---|
| fork | CreateProcess | Create a new process |
| waitpid | WaitForSingleObject | Can wait for a process to exit |
| execve | (none) | CreateProcess = fork + execve |
| exit | ExitProcess | Terminate execution |
| open | CreateFile | Create a file or open an existing file |
| close | CloseHandle | Close a file |
| read | ReadFile | Read data from a file |
| write | WriteFile | Write data to a file |
| lseek | SetFilePointer | Move the file pointer |
| stat | GetFileAttributesEx | Get various file attributes |
| mkdir | CreateDirectory | Create a new directory |
| rmdir | RemoveDirectory | Remove an empty directory |
| link | (none) | Win32 does not support links |
| unlink | DeleteFile | Destroy an existing file |
| mount | (none) | Win32 does not support mount |
| umount | (none) | Win32 does not support mount |
| chdir | SetCurrentDirectory | Change the current working directory |
| chmod | (none) | Win32 does not support security (although NT does) |
| kill | (none) | Win32 does not support signals |
| time | GetLocalTime | Get the current time |

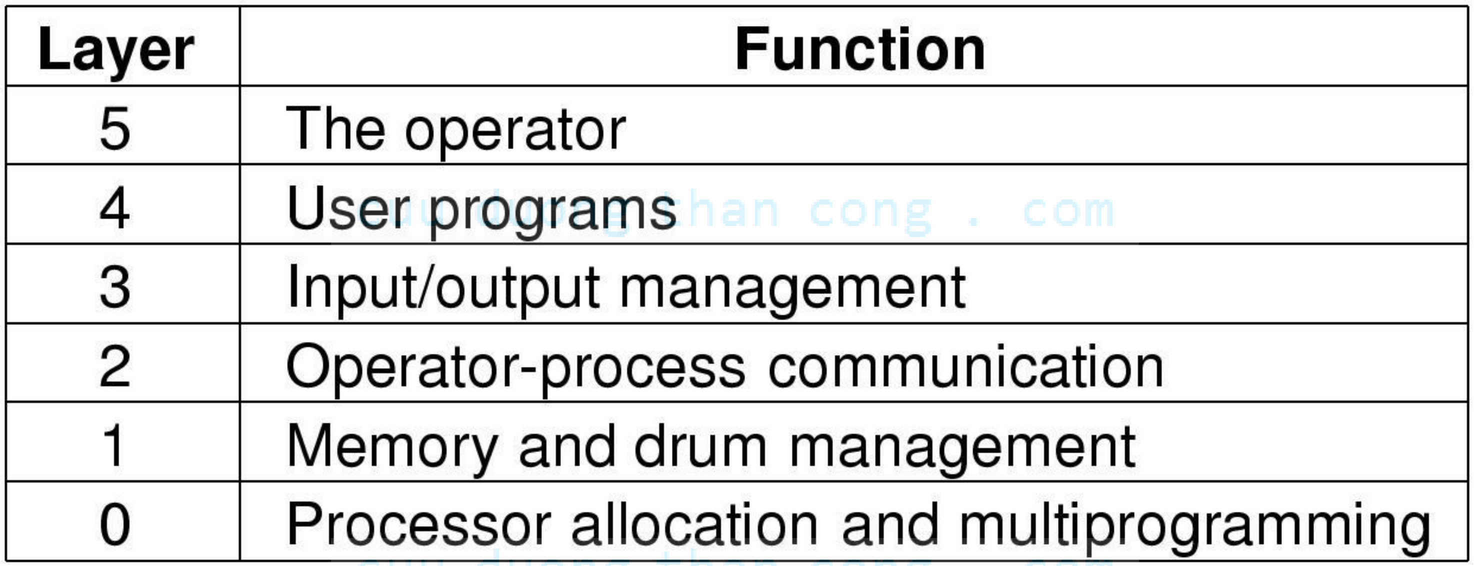
## Some Win32 API calls

# Operating System Structure (1)



Simple structuring model for a monolithic system

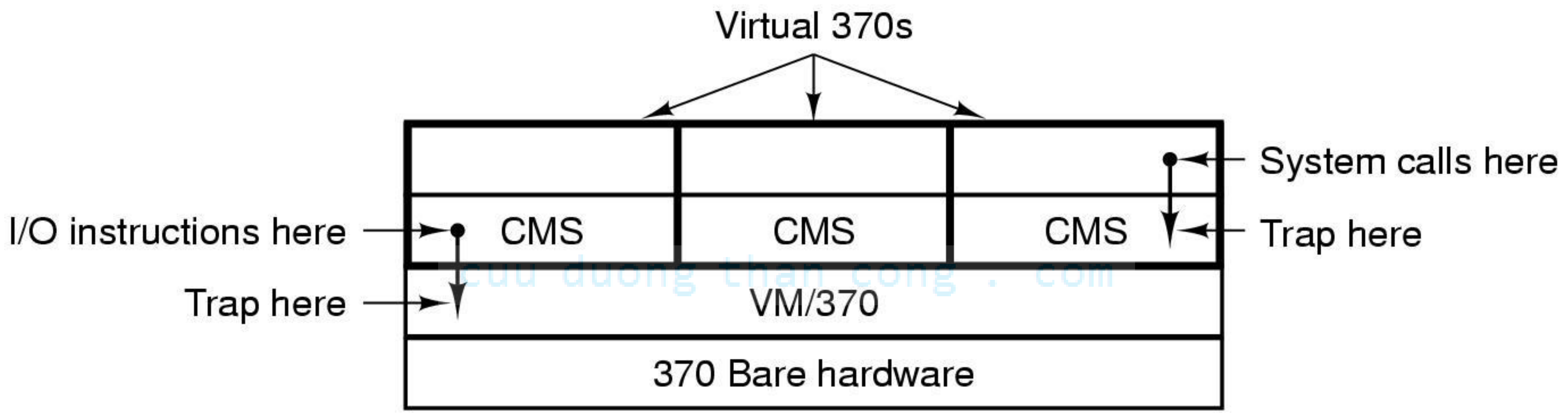
# Operating System Structure (2)



| Layer | Function |
|---|---|
| 5 | The operator |
| 4 | User programs |
| 3 | Input/output management |
| 2 | Operator-process communication |
| 1 | Memory and drum management |
| 0 | Processor allocation and multiprogramming |

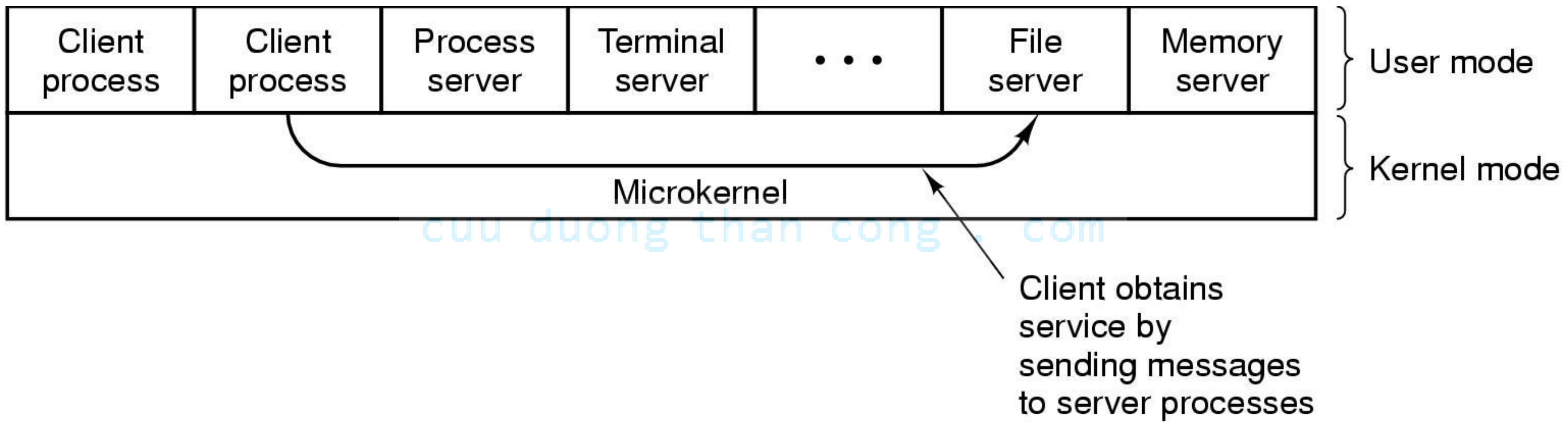
Structure of the THE operating system

# Operating System Structure (3)



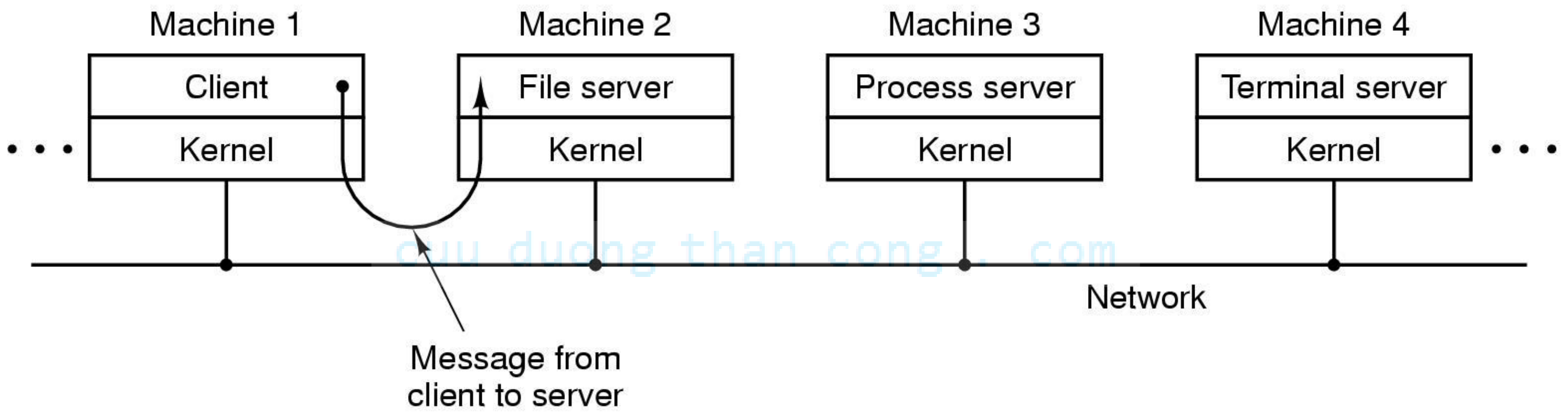
Structure of VM/370 with CMS

# Operating System Structure (4)



The client-server model

## Operating System Structure (5)



The client-server model in a distributed system

# Metric Units

| Exp. | Explicit | Prefix | Exp. | Explicit | Prefix |
|---|---|---|---|---|---|
| $10^{-3}$ | 0.001 | milli | $10^{3}$ | 1,000 | Kilo |
| $10^{-6}$ | 0.000001 | micro | $10^{6}$ | 1,000,000 | Mega |
| $10^{-9}$ | 0.000000001 | nano | $10^{9}$ | 1,000,000,000 | Giga |
| $10^{-12}$ | 0.000000000001 | pico | $10^{12}$ | 1,000,000,000,000 | Tera |
| $10^{-15}$ | 0.000000000000001 | femto | $10^{15}$ | 1,000,000,000,000,000 | Peta |
| $10^{-18}$ | 0.000000000000000001 | atto | $10^{18}$ | 1,000,000,000,000,000,000 | Exa |
| $10^{-21}$ | 0.000000000000000000001 | zepto | $10^{21}$ | 1,000,000,000,000,000,000,000 | Zetta |
| $10^{-24}$ | 0.000000000000000000000001 | yocto | $10^{24}$ | 1,000,000,000,000,000,000,000,000 | Yotta |

The metric prefixes