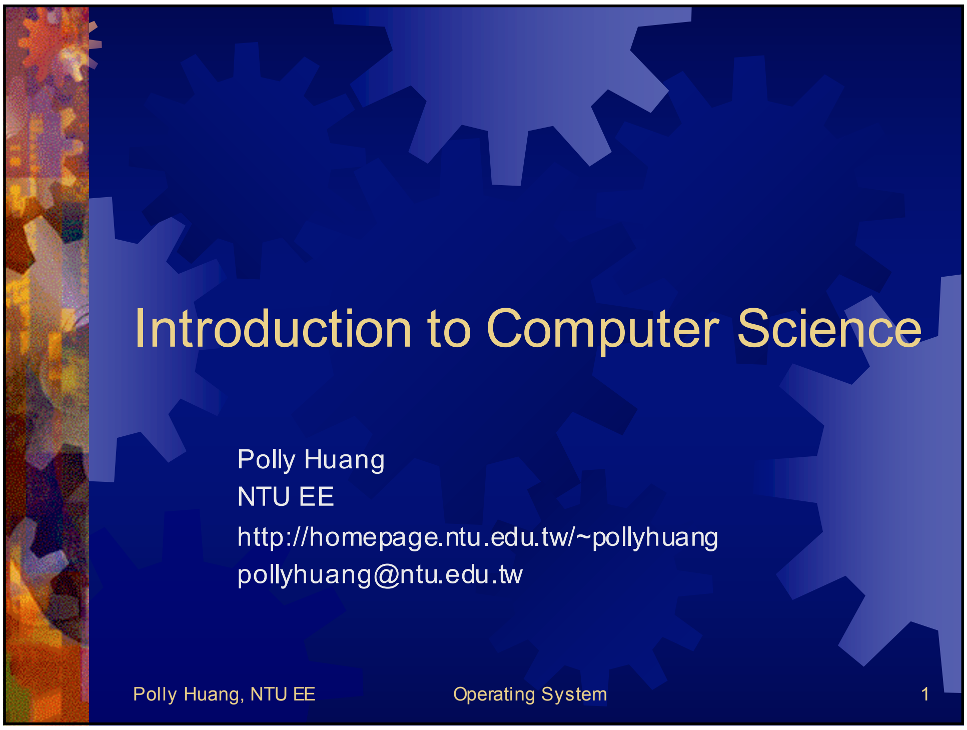
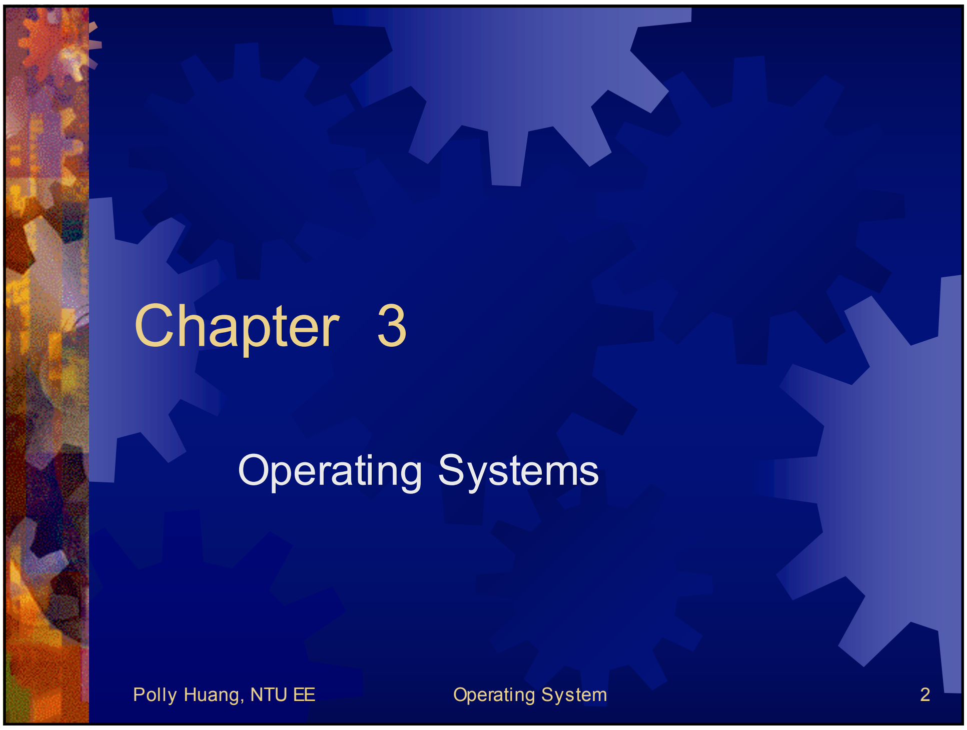


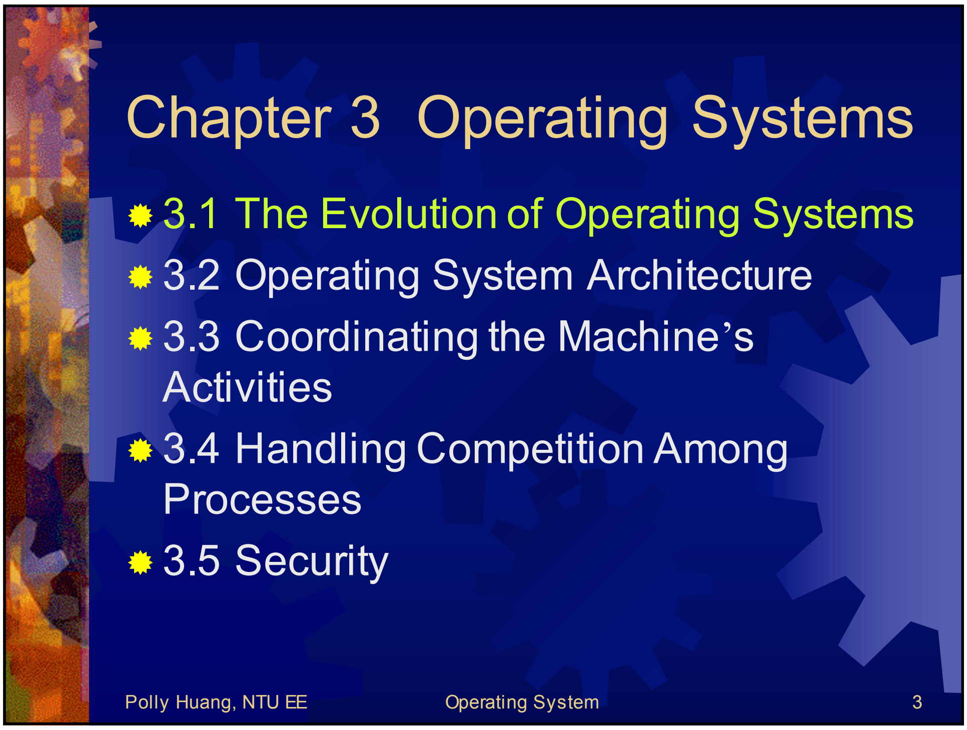
Introduction to Computer Science

Polly Huang
NTU EE
http://homepage.ntu.edu.tw/~pollyhuang
pollyhuang@ntu.edu.tw



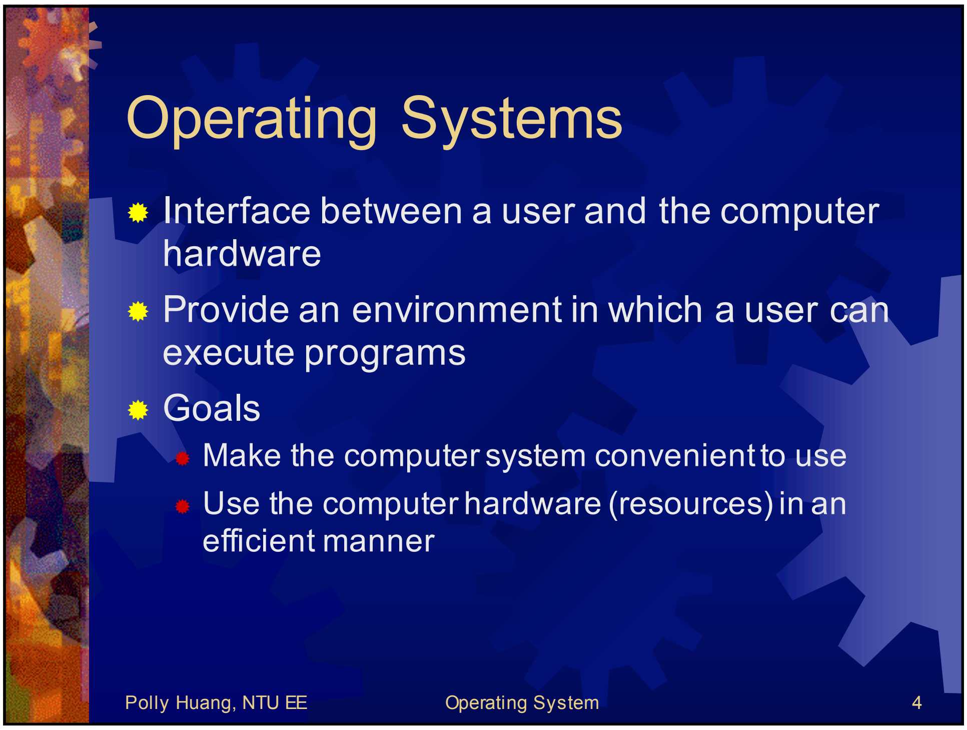
Chapter 3

Operating Systems



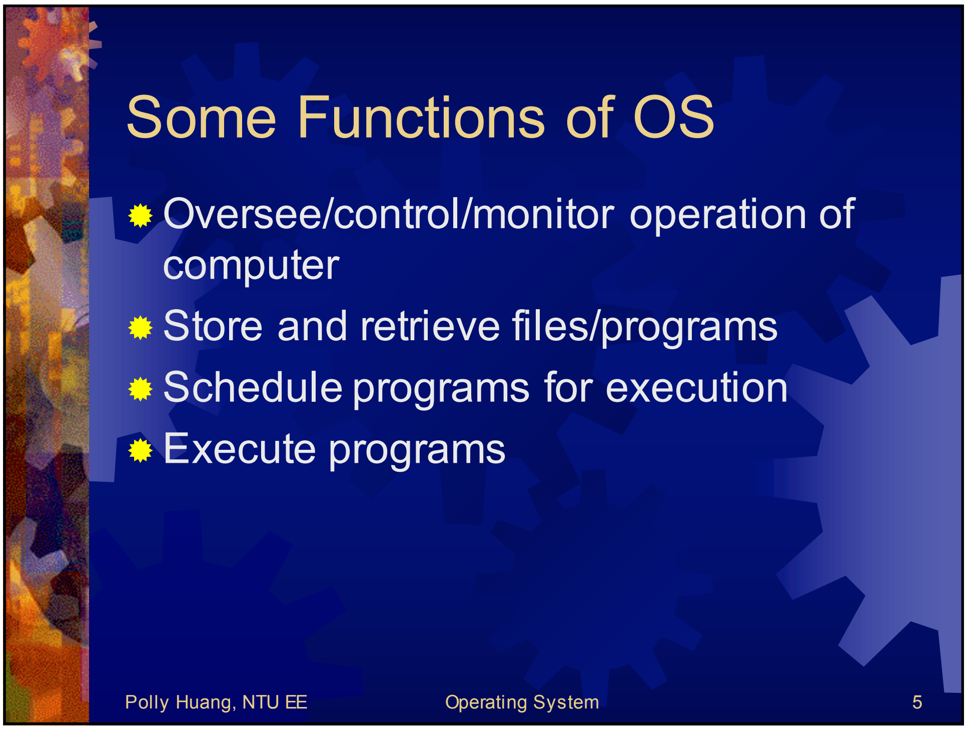
Chapter 3 Operating Systems

- 3.1 The Evolution of Operating Systems
- 3.2 Operating System Architecture
- 3.3 Coordinating the Machine's Activities
- 3.4 Handling Competition Among Processes
- 3.5 Security



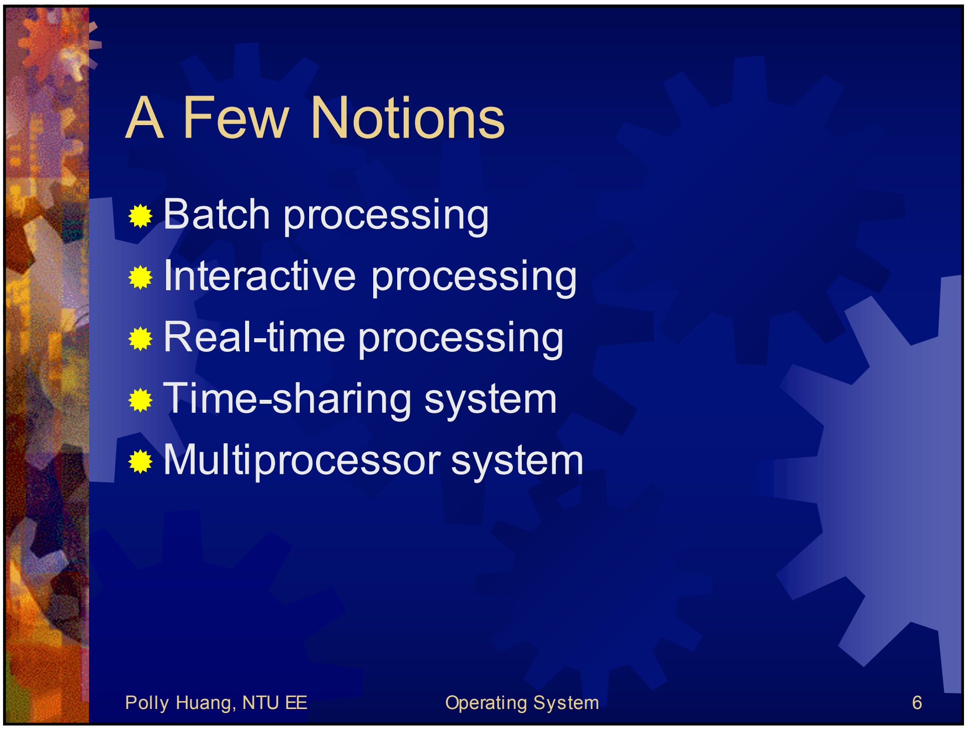
Operating Systems

- Interface between a user and the computer hardware
- Provide an environment in which a user can execute programs
- Goals
  - Make the computer system convenient to use
  - Use the computer hardware (resources) in an efficient manner



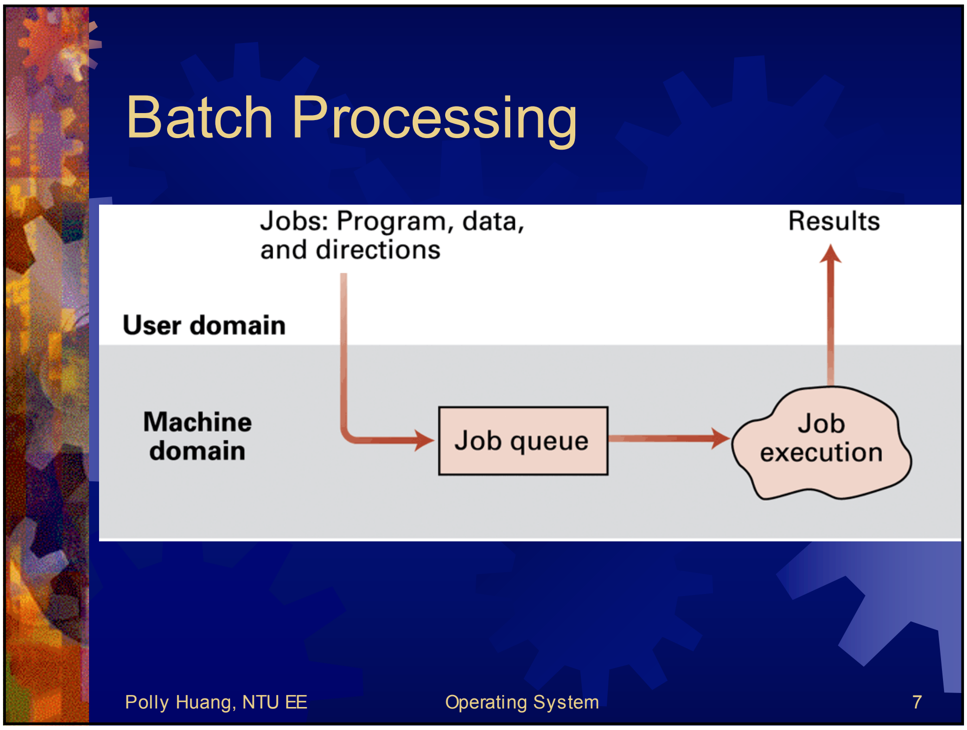
Some Functions of OS

- Oversee/control/monitor operation of computer
- Store and retrieve files/programs
- Schedule programs for execution
- Execute programs



A Few Notions

- Batch processing
- Interactive processing
- Real-time processing
- Time-sharing system
- Multiprocessor system



Batch Processing

Jobs: Program, data, and directions

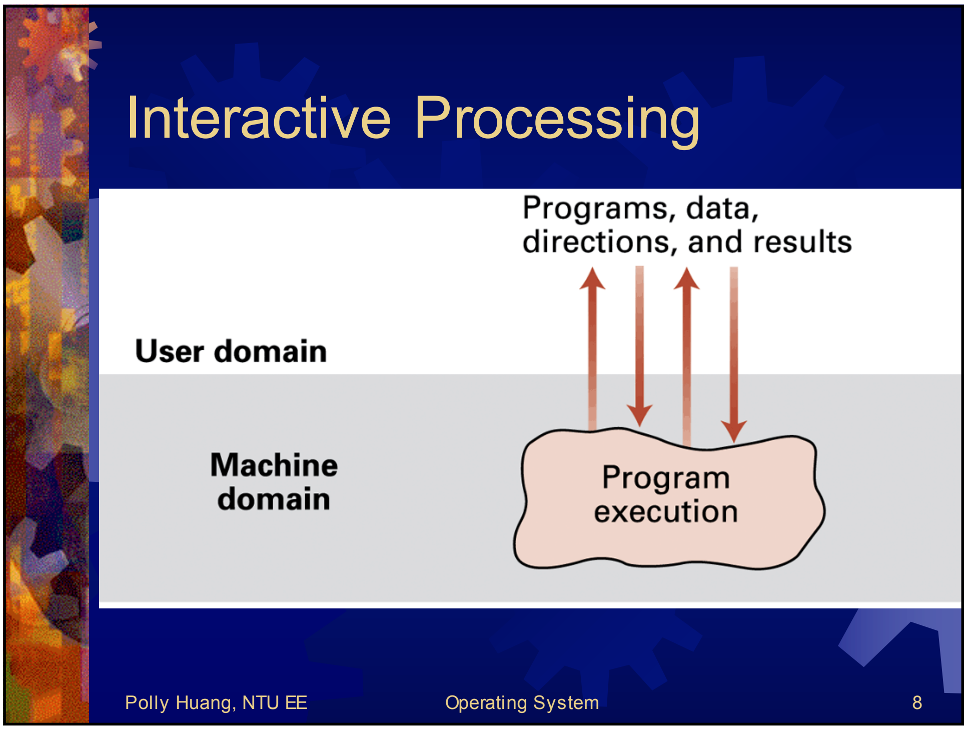
Results

User domain

Machine domain

Job queue

Job execution



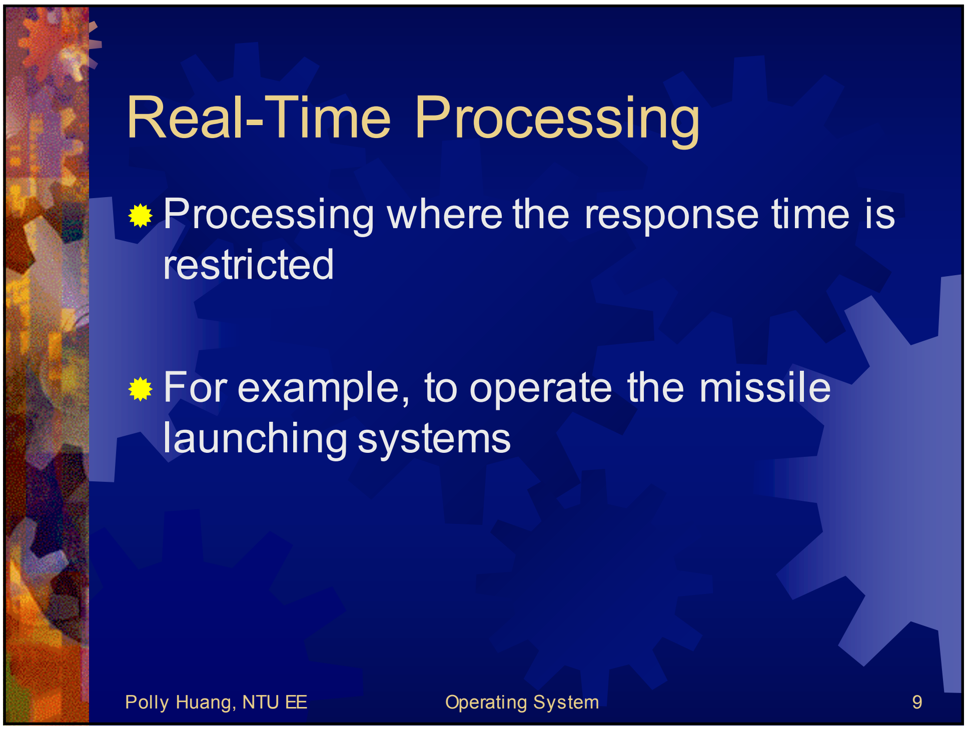
Interactive Processing

Programs, data, directions, and results

User domain

Machine domain

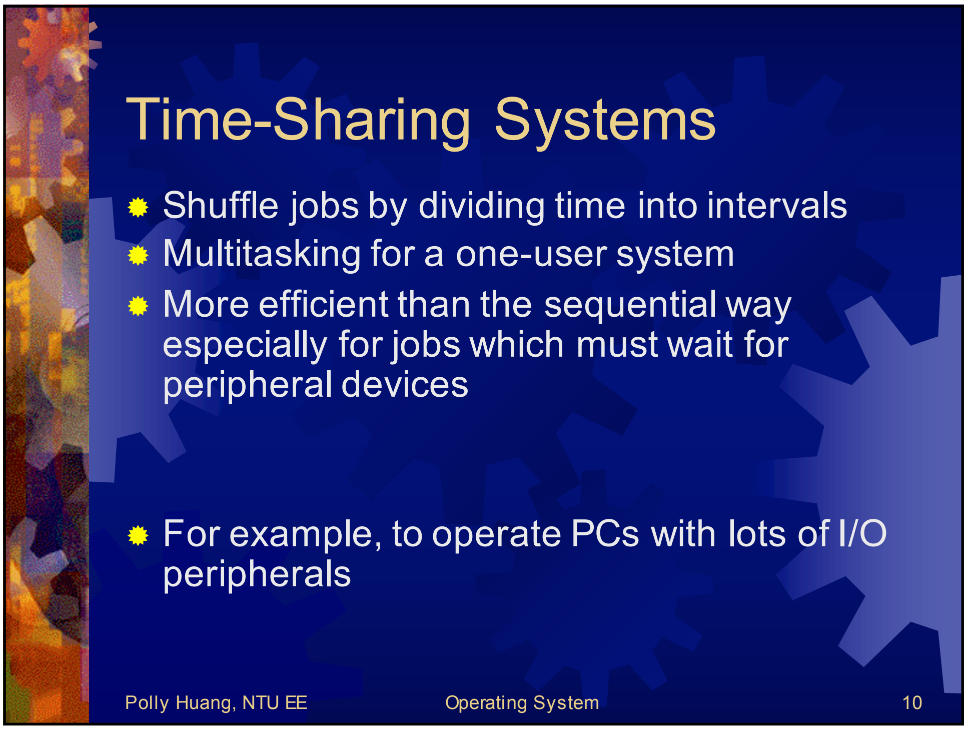
Program execution



Real-Time Processing

- Processing where the response time is restricted
- For example, to operate the missile launching systems

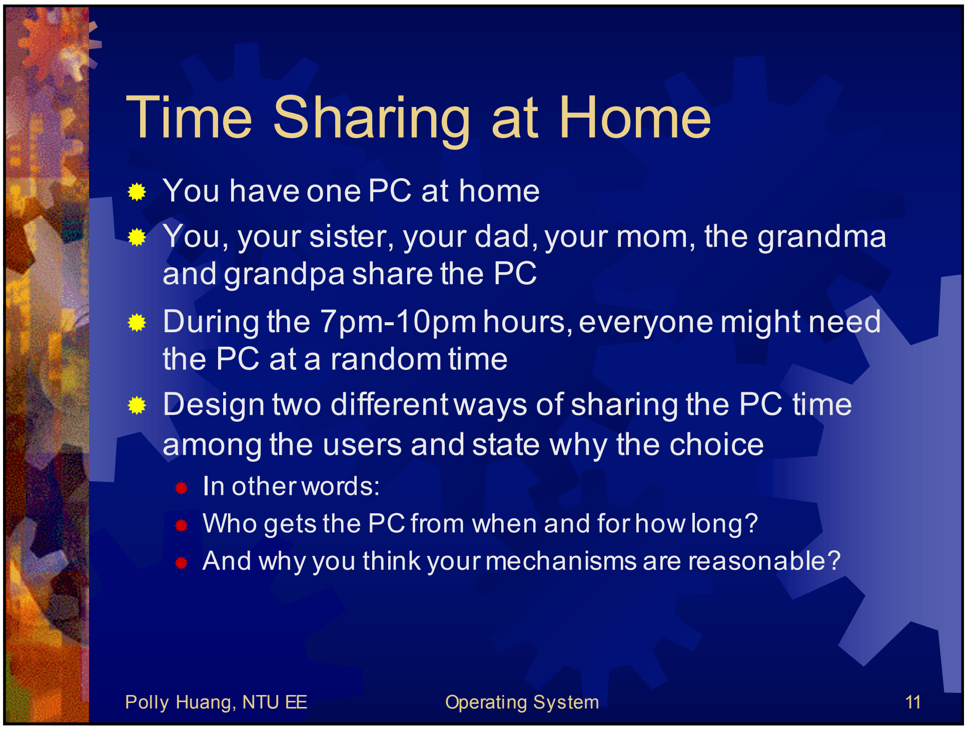
Polly Huang, NTU EE Operating System 9



Time-Sharing Systems

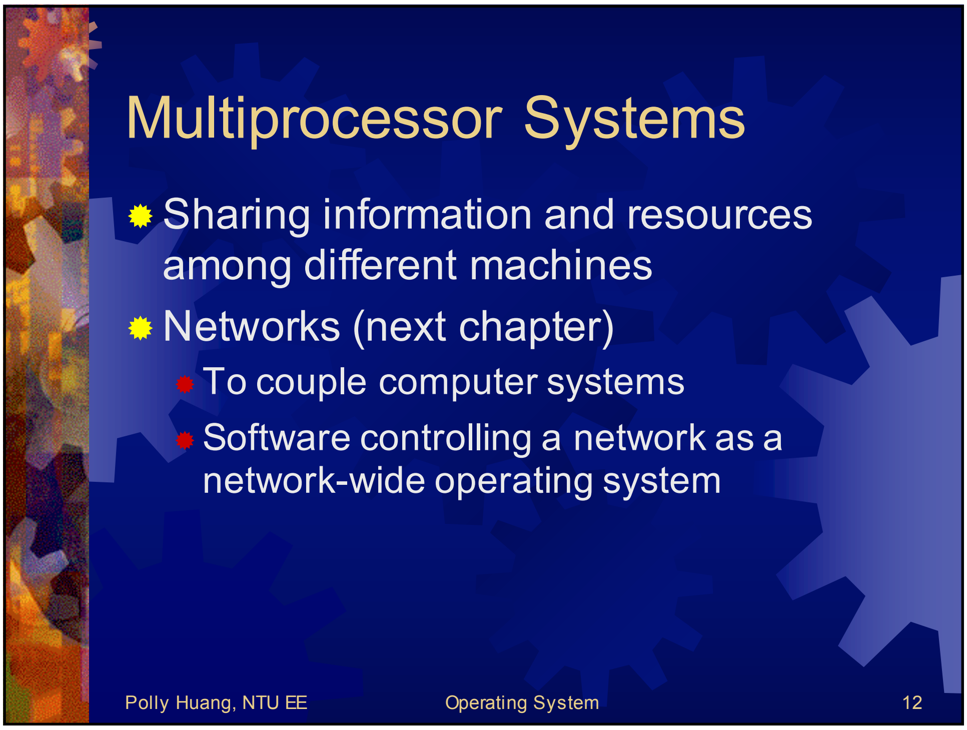
- Shuffle jobs by dividing time into intervals
- Multitasking for a one-user system
- More efficient than the sequential way especially for jobs which must wait for peripheral devices
- For example, to operate PCs with lots of I/O peripherals

Polly Huang, NTU EE Operating System 10



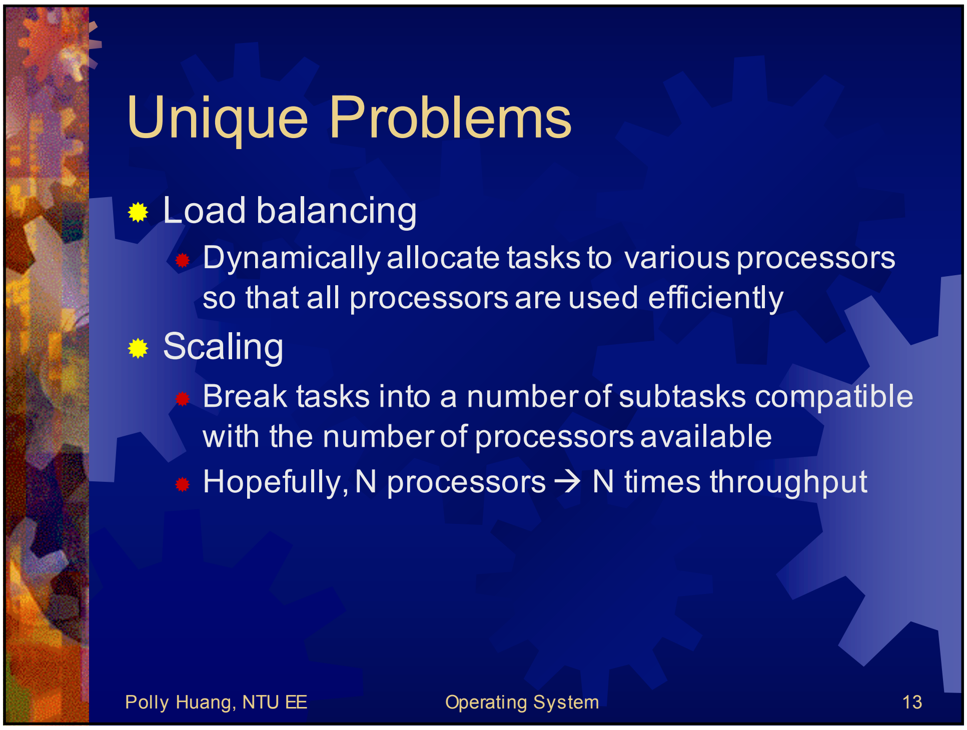
Time Sharing at Home

- You have one PC at home
- You, your sister, your dad, your mom, the grandma and grandpa share the PC
- During the 7pm-10pm hours, everyone might need the PC at a random time
- Design two different ways of sharing the PC time among the users and state why the choice
  - In other words:
  - Who gets the PC from when and for how long?
  - And why you think your mechanisms are reasonable?



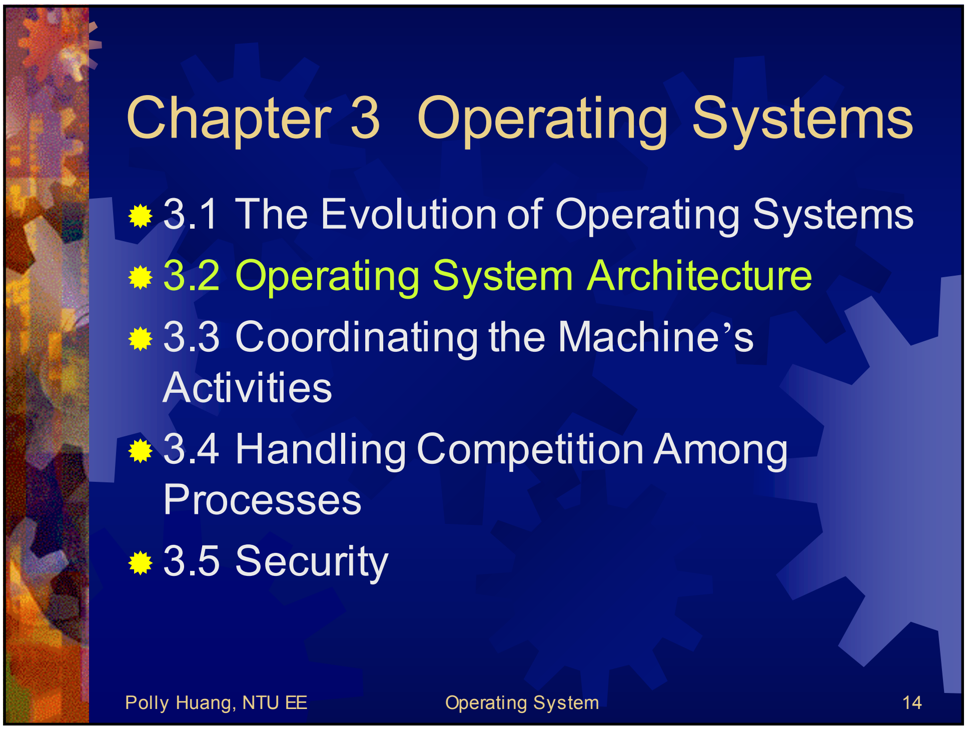
Multiprocessor Systems

- Sharing information and resources among different machines
- Networks (next chapter)
  - To couple computer systems
  - Software controlling a network as a network-wide operating system



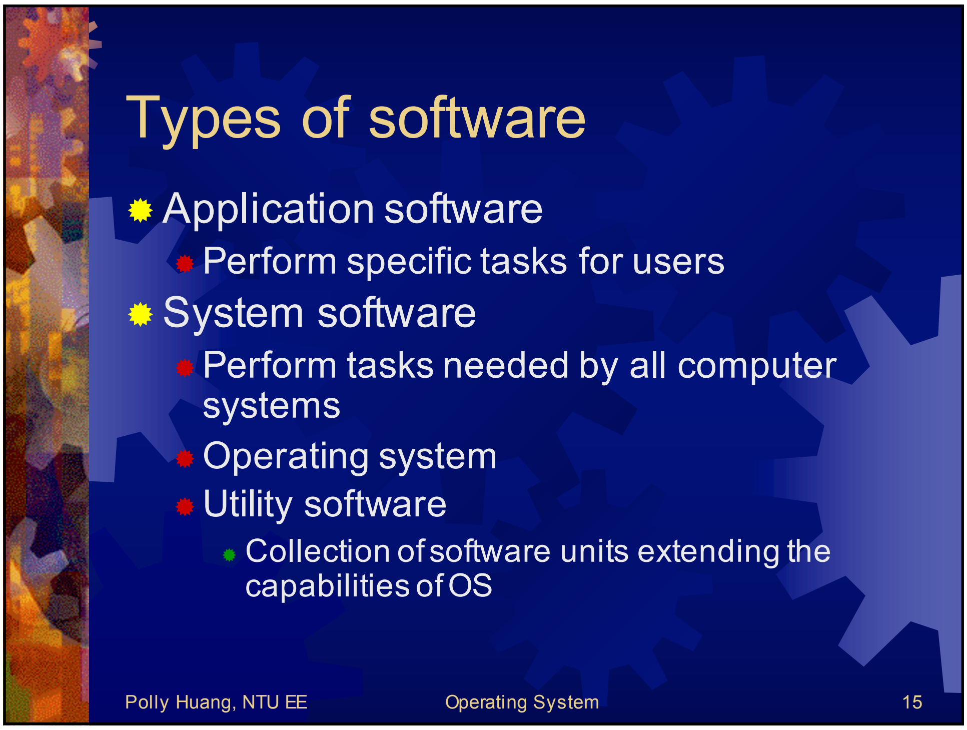
Unique Problems

- Load balancing
  - Dynamically allocate tasks to various processors so that all processors are used efficiently
- Scaling
  - Break tasks into a number of subtasks compatible with the number of processors available
  - Hopefully, N processors → N times throughput



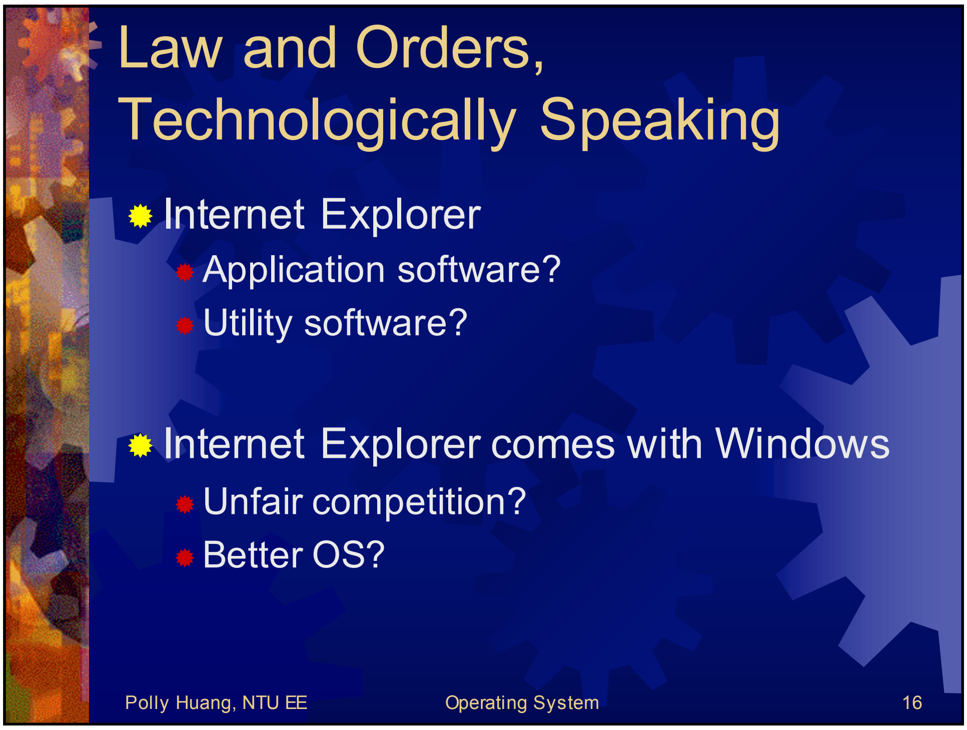
Chapter 3 Operating Systems

- 3.1 The Evolution of Operating Systems
- 3.2 Operating System Architecture
- 3.3 Coordinating the Machine's Activities
- 3.4 Handling Competition Among Processes
- 3.5 Security



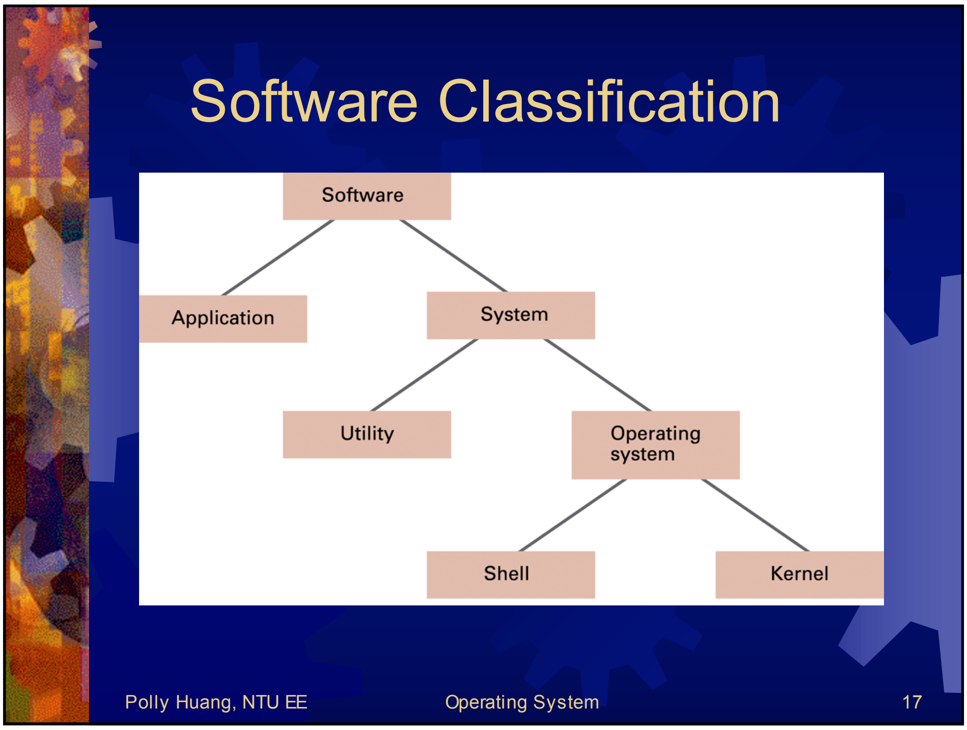
Types of software

- Application software
  - Perform specific tasks for users
- System software
  - Perform tasks needed by all computer systems
  - Operating system
  - Utility software
    - Collection of software units extending the capabilities of OS



Law and Orders, Technologically Speaking

- Internet Explorer
  - Application software?
  - Utility software?
- Internet Explorer comes with Windows
  - Unfair competition?
  - Better OS?



Software Classification

Software

Application

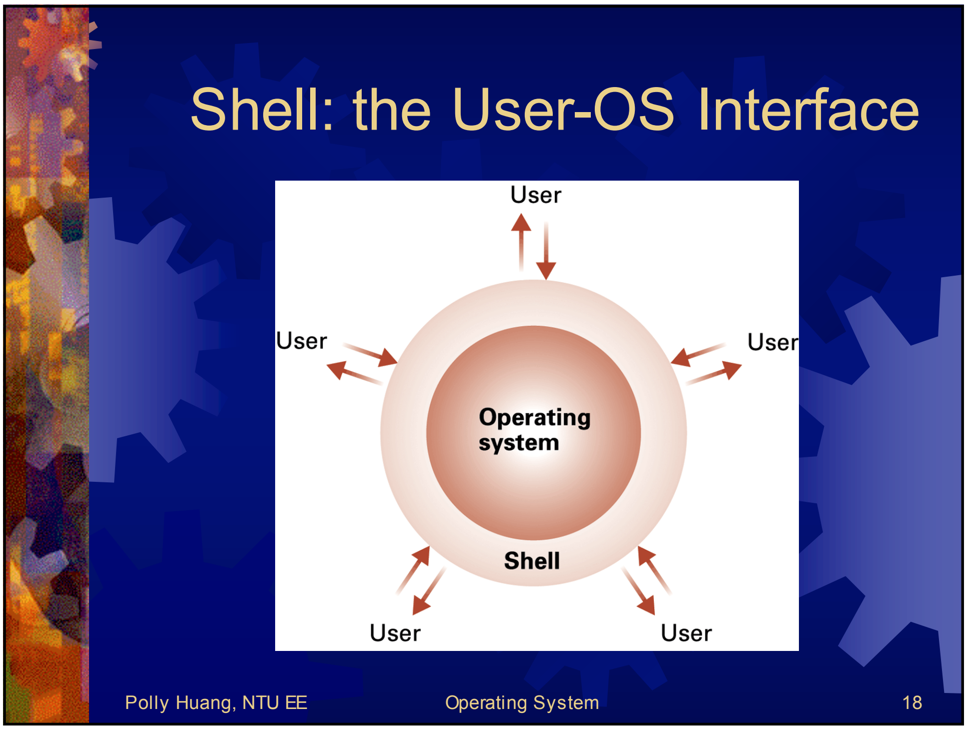
System

Utility

Operating system

Shell

Kernel



Shell: the User-OS Interface

User

User

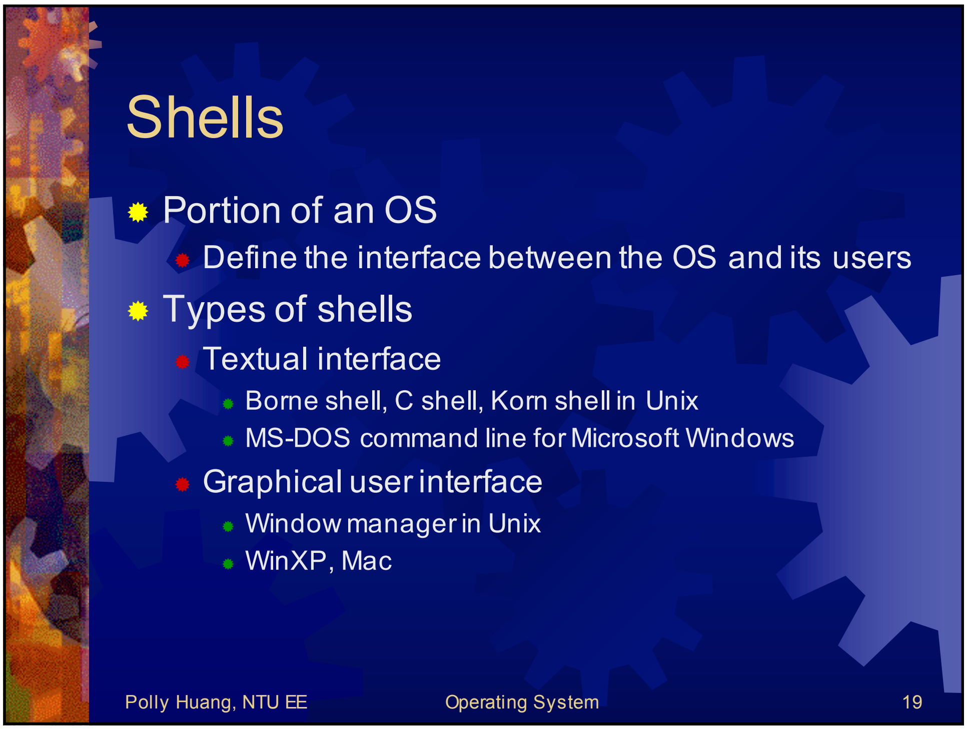
User

Operating system

Shell

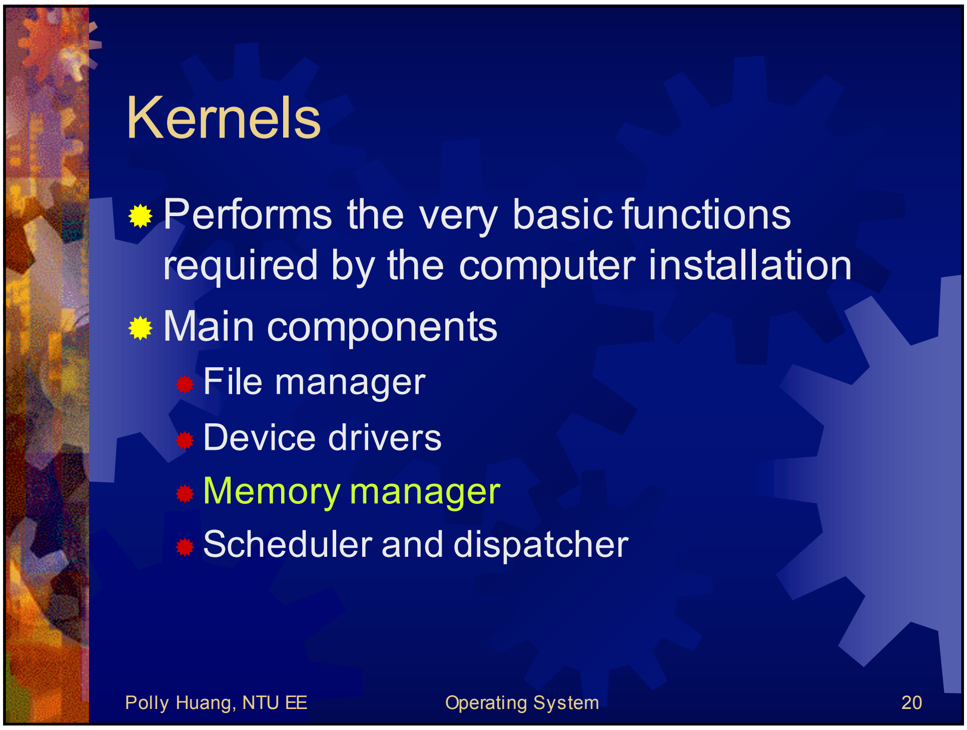
User

User



Shells

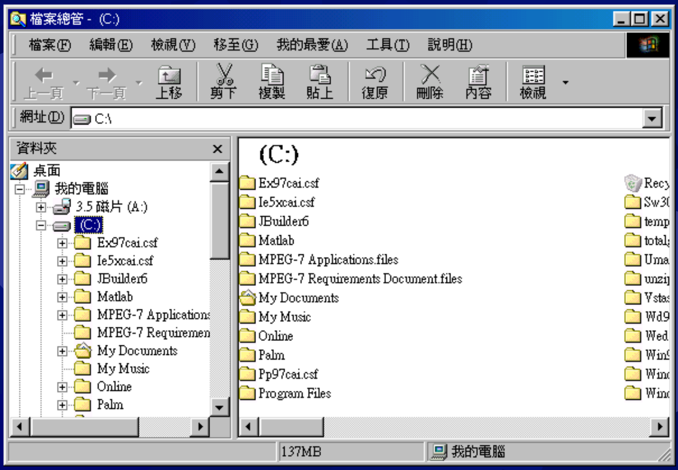
- Portion of an OS
  - Define the interface between the OS and its users
- Types of shells
  - Textual interface
    - Borne shell, C shell, Korn shell in Unix
    - MS-DOS command line for Microsoft Windows
  - Graphical user interface
    - Window manager in Unix
    - WinXP, Mac



Kernels

- Performs the very basic functions required by the computer installation
- Main components
  - File manager
  - Device drivers
  - Memory manager
  - Scheduler and dispatcher

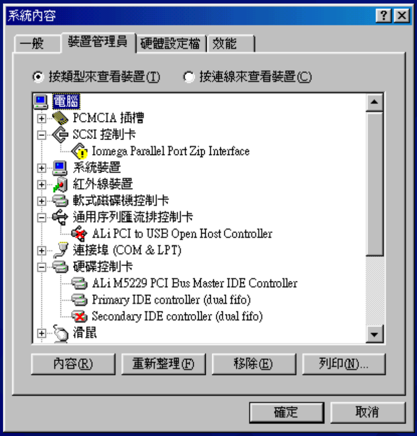
File Manager



File System

- Directory or folder
  - User-created group or bundle of files
- Path: position of a file in directory hierarchy
  - *C:\animals\prehistoric\dinosaurs*
- Any access to a file by other software units is obtained with the help of the file manager
- File descriptor: information needed to access an open file

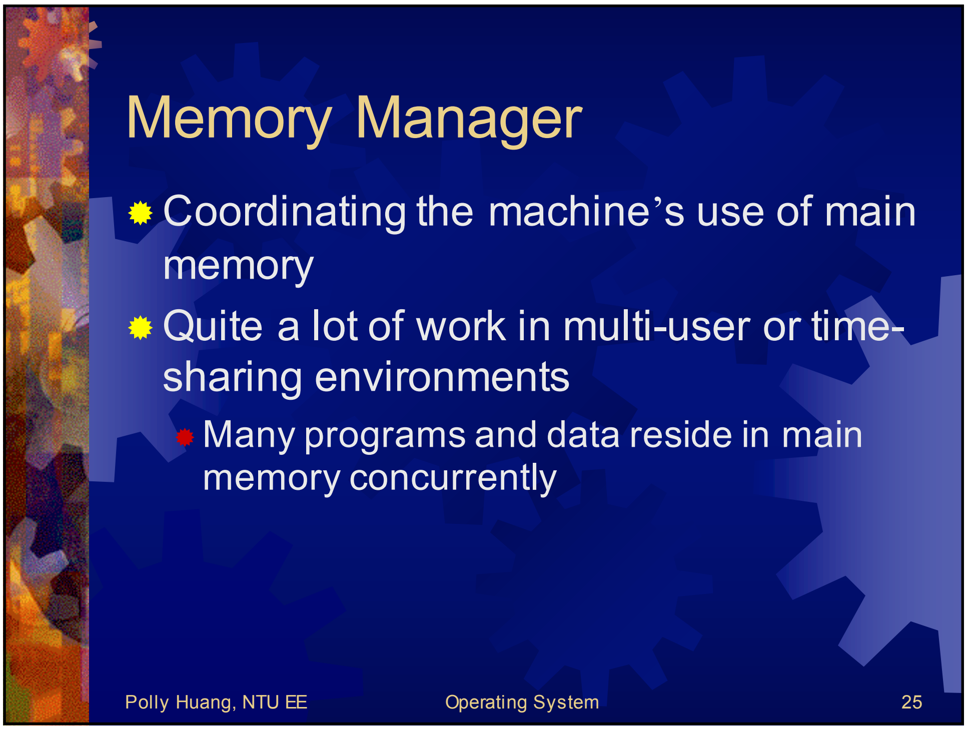
Device Manager



● 按類型來查看裝置(T) ○ 按連線來查看裝置(C)

Device Driver

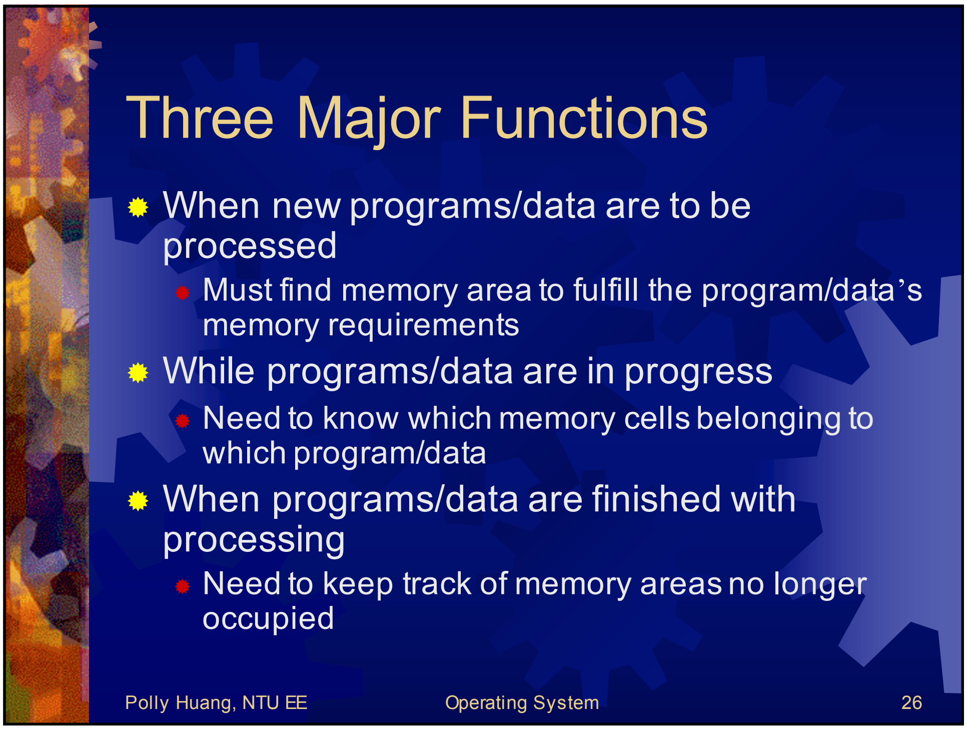
- Software units that
  - 1. tightly coupled with the controllers
  - 2. to carry out operations on the peripheral devices attached to the machine
- Each device driver is uniquely designed for its particular type of device
  - Example: Device driver for a printer contains software to read and decode status word as well as other handshaking details. Other software does not have to deal with those details in order to print a file



Memory Manager

- Coordinating the machine’s use of main memory
- Quite a lot of work in multi-user or time-sharing environments
  - Many programs and data reside in main memory concurrently

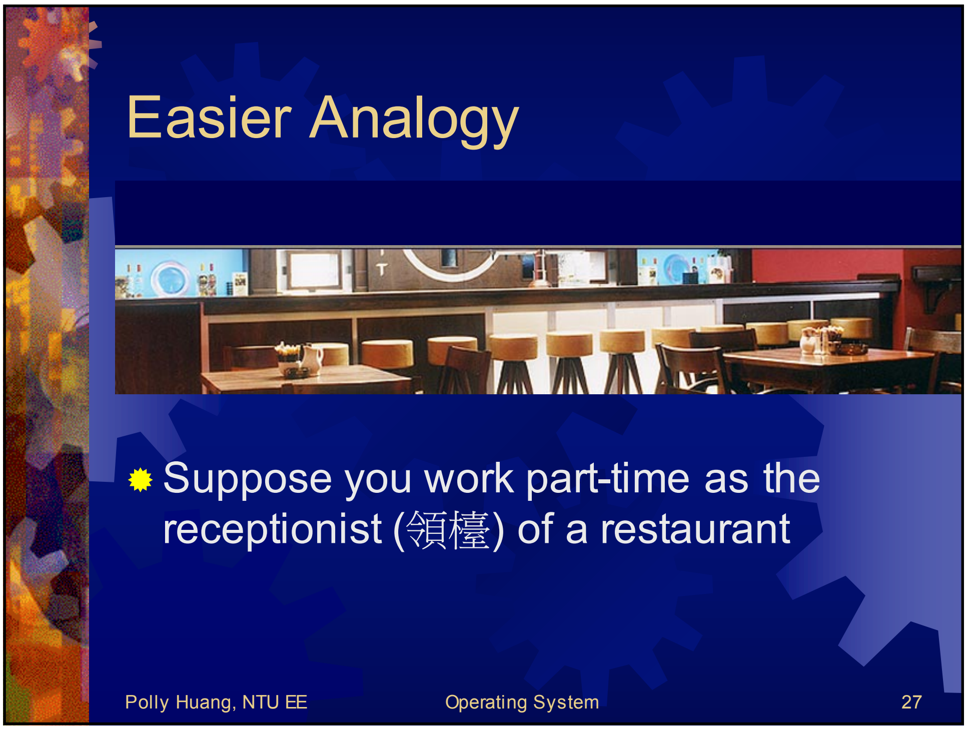
Polly Huang, NTU EE Operating System 25



Three Major Functions

- When new programs/data are to be processed
  - Must find memory area to fulfill the program/data’s memory requirements
- While programs/data are in progress
  - Need to know which memory cells belonging to which program/data
- When programs/data are finished with processing
  - Need to keep track of memory areas no longer occupied

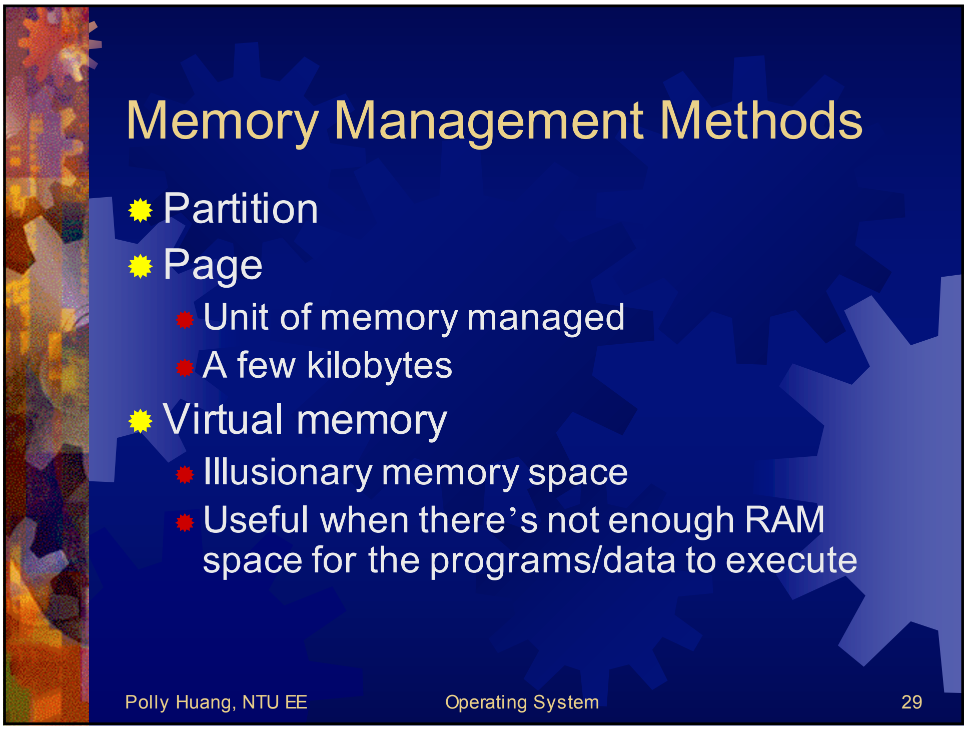
Polly Huang, NTU EE Operating System 26



Easier Analogy

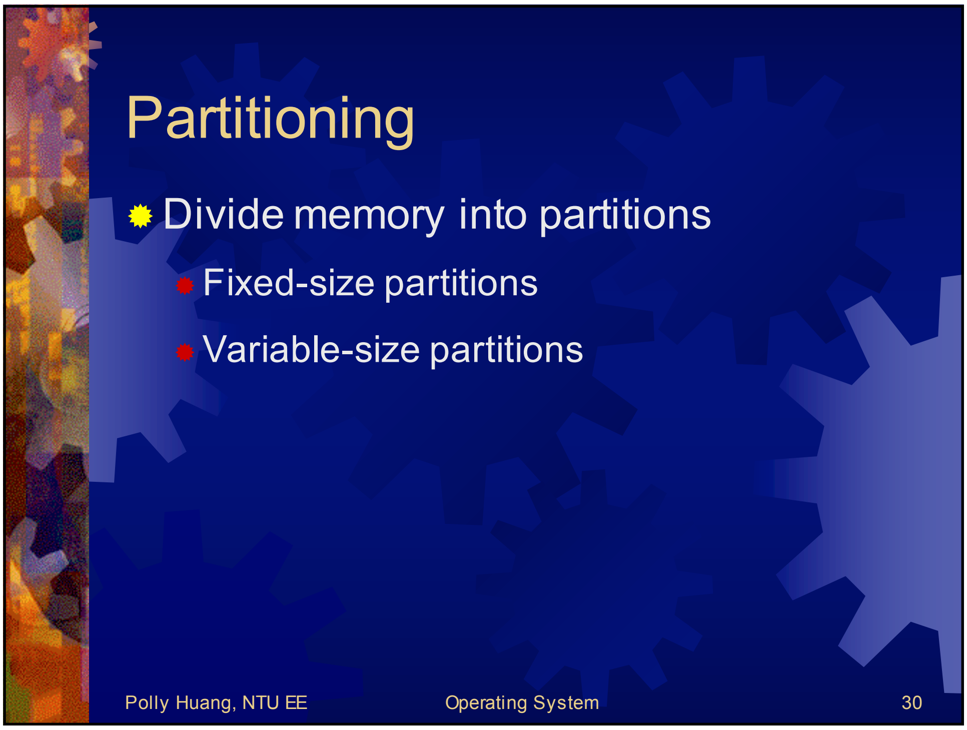
- Suppose you work part-time as the receptionist (領檯) of a restaurant

Quiz Time!



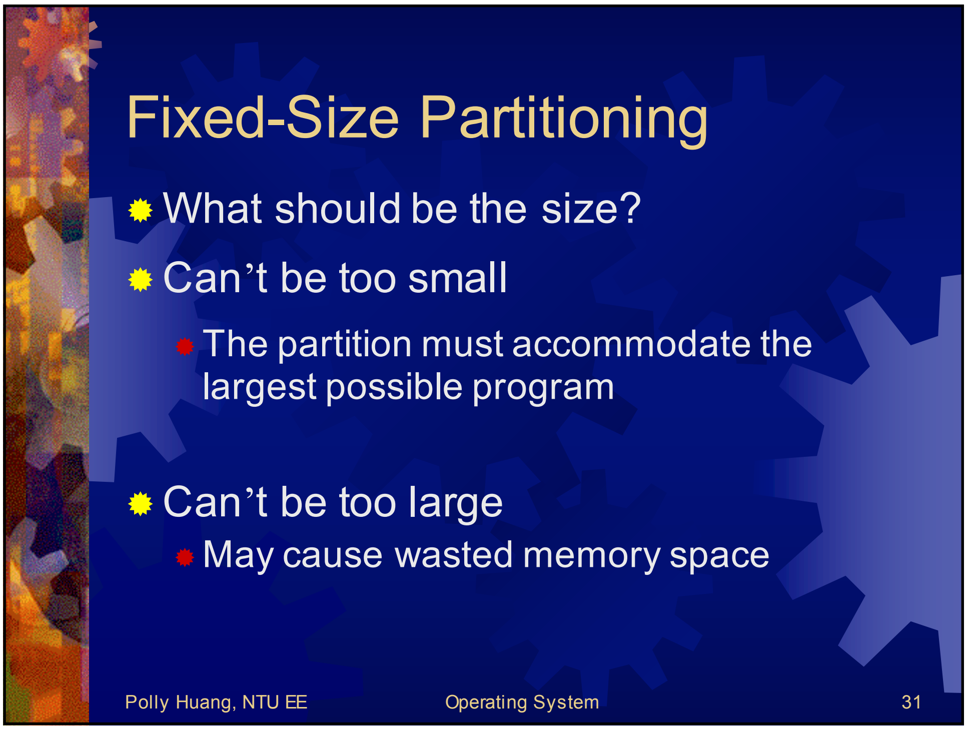
Memory Management Methods

- Partition
- Page
  - Unit of memory managed
  - A few kilobytes
- Virtual memory
  - Illusionary memory space
  - Useful when there's not enough RAM space for the programs/data to execute



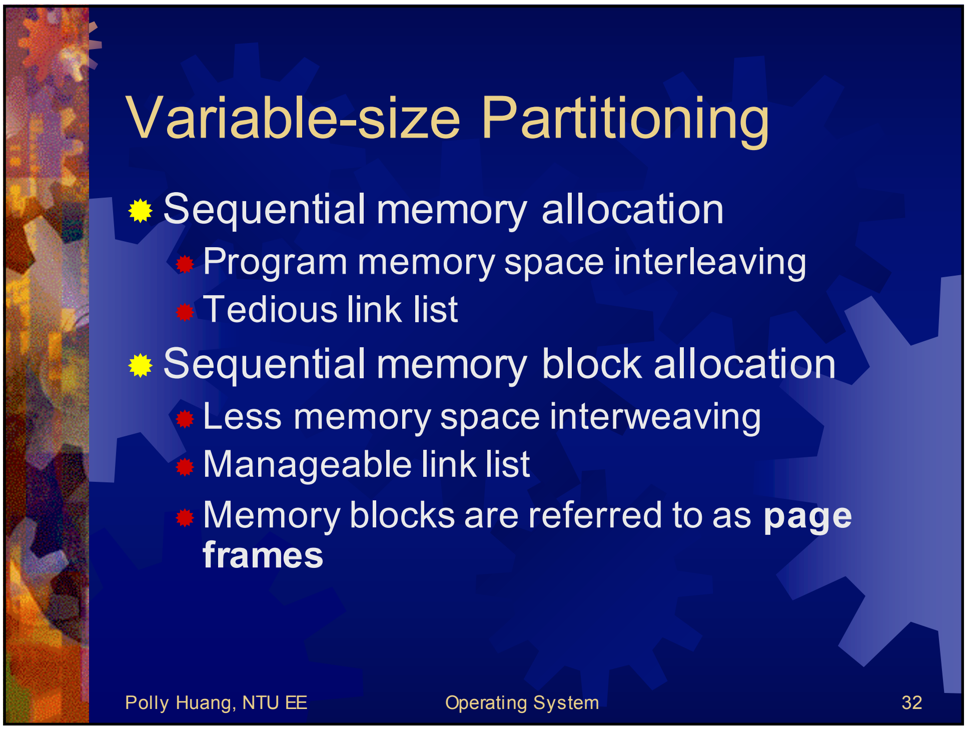
Partitioning

- Divide memory into partitions
  - Fixed-size partitions
  - Variable-size partitions



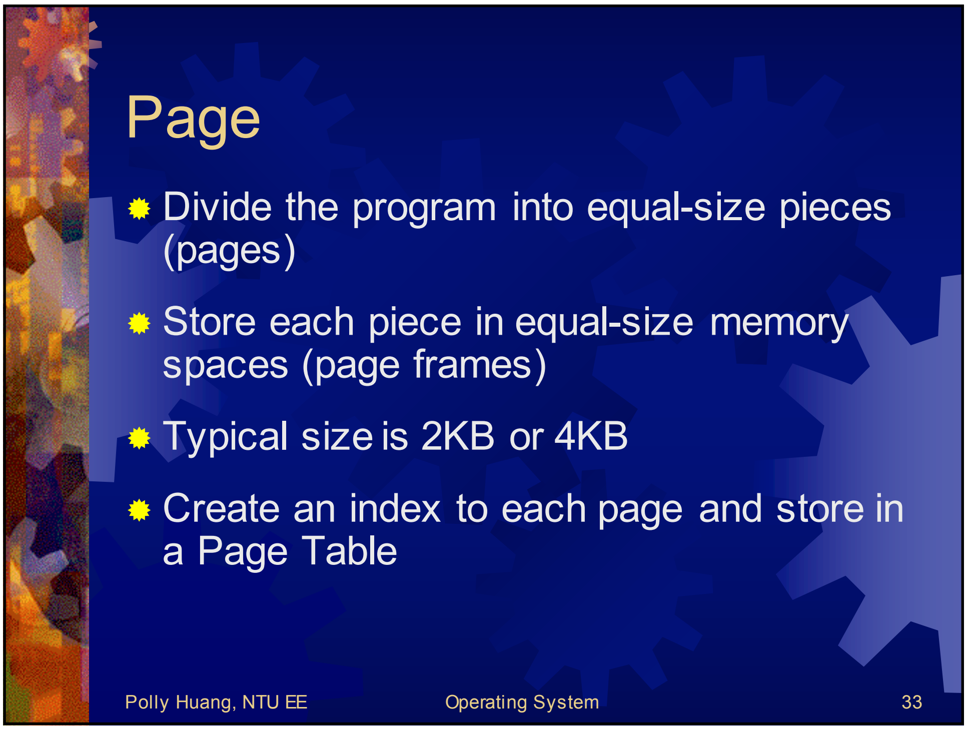
Fixed-Size Partitioning

- What should be the size?
- Can’t be too small
  - The partition must accommodate the largest possible program
- Can’t be too large
  - May cause wasted memory space



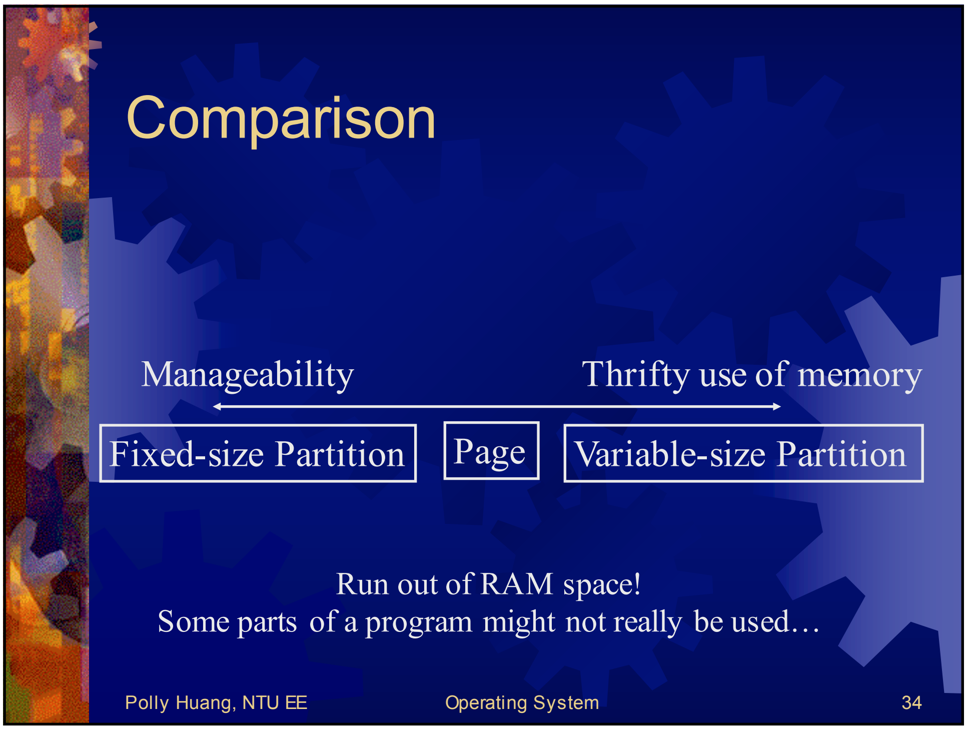
Variable-size Partitioning

- Sequential memory allocation
  - Program memory space interleaving
  - Tedious link list
- Sequential memory block allocation
  - Less memory space interweaving
  - Manageable link list
  - Memory blocks are referred to as **page frames**



Page

- Divide the program into equal-size pieces (pages)
- Store each piece in equal-size memory spaces (page frames)
- Typical size is 2KB or 4KB
- Create an index to each page and store in a Page Table

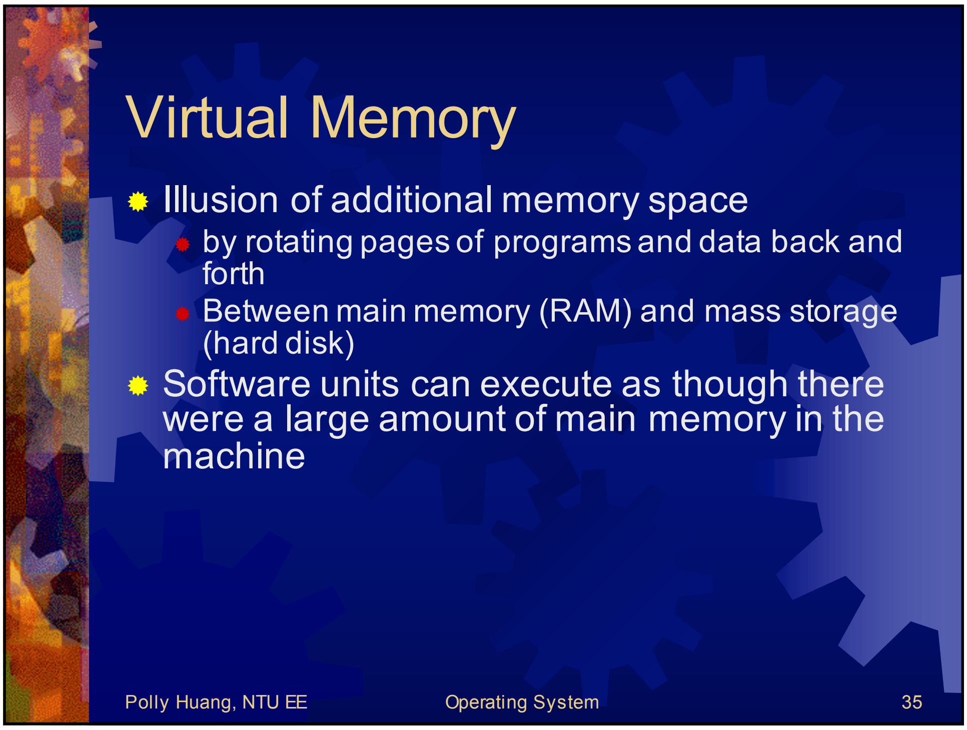


Comparison

Manageability ⟷ Thrifty use of memory

Fixed-size Partition | Page | Variable-size Partition

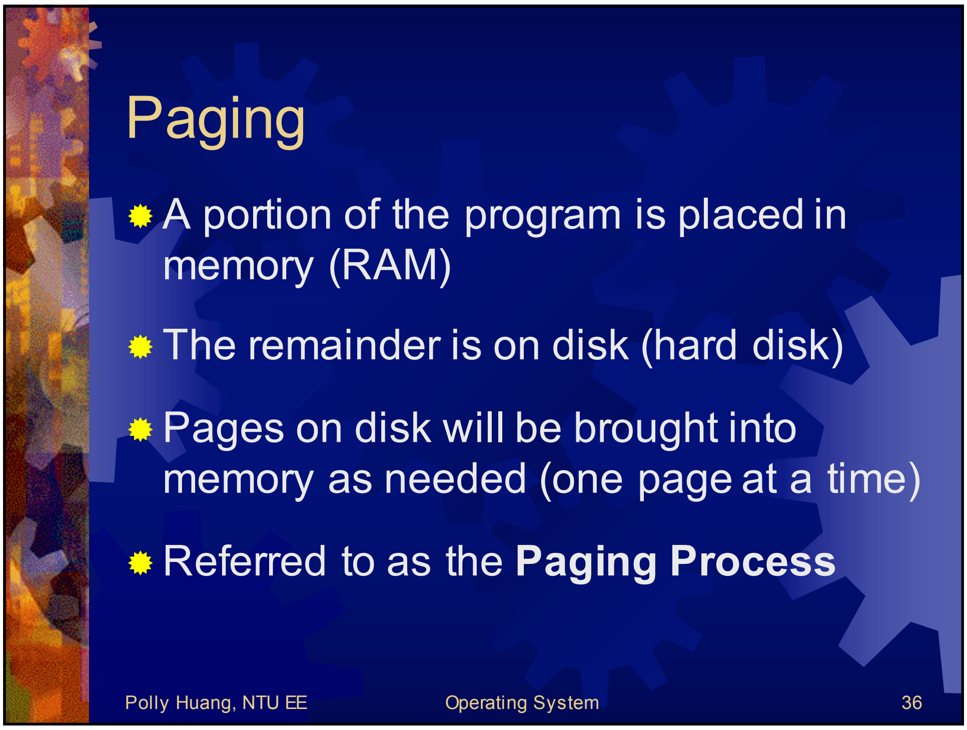
Run out of RAM space!
Some parts of a program might not really be used…



Virtual Memory

- Illusion of additional memory space
  - by rotating pages of programs and data back and forth
  - Between main memory (RAM) and mass storage (hard disk)
- Software units can execute as though there were a large amount of main memory in the machine

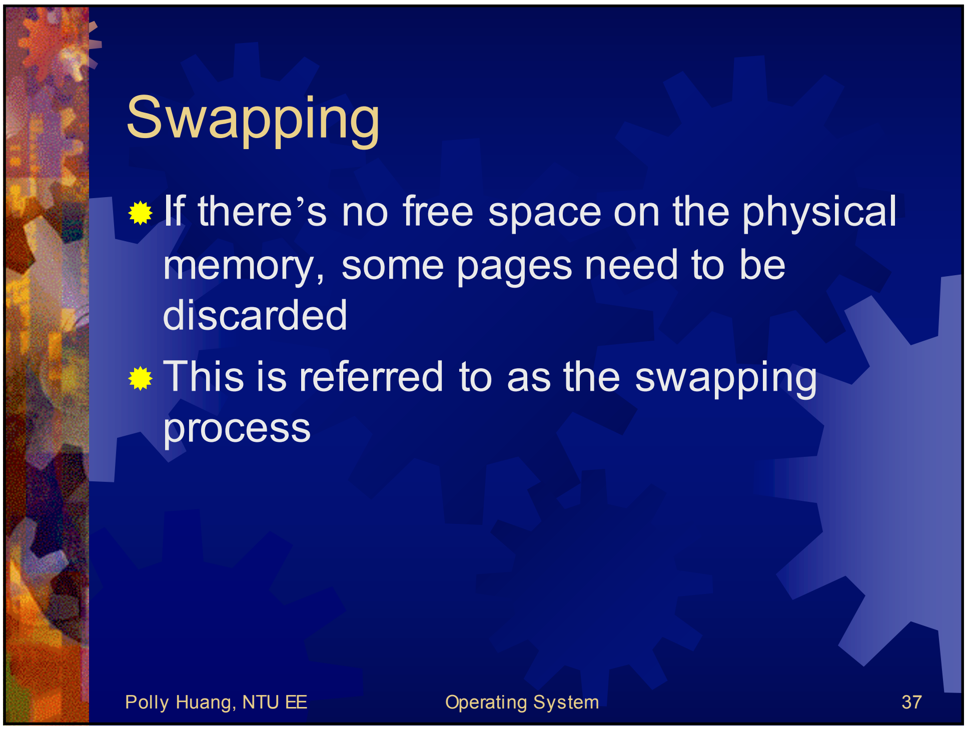
Polly Huang, NTU EE Operating System 35



Paging

- A portion of the program is placed in memory (RAM)
- The remainder is on disk (hard disk)
- Pages on disk will be brought into memory as needed (one page at a time)
- Referred to as the **Paging Process**

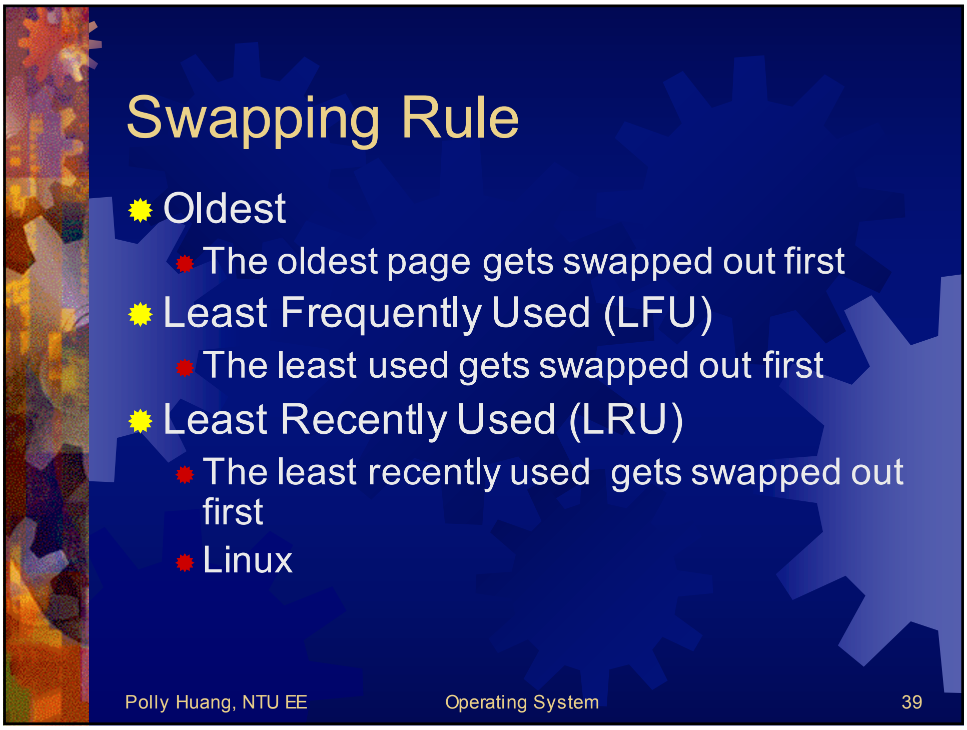
Polly Huang, NTU EE Operating System 36



Swapping

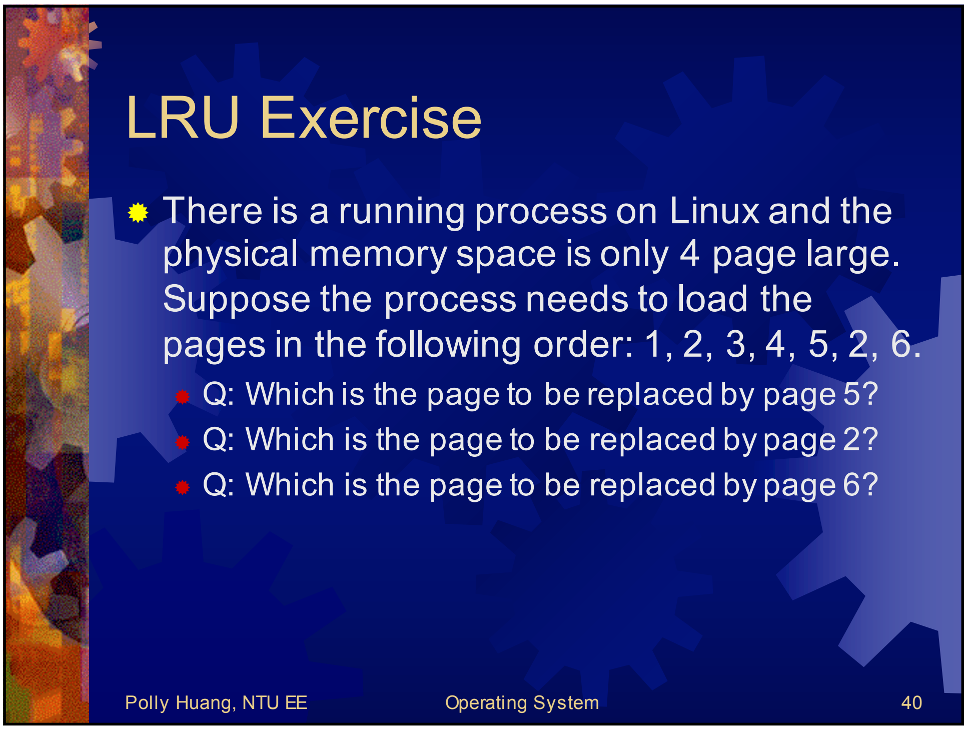
- If there’s no free space on the physical memory, some pages need to be discarded
- This is referred to as the swapping process

Quiz Time!



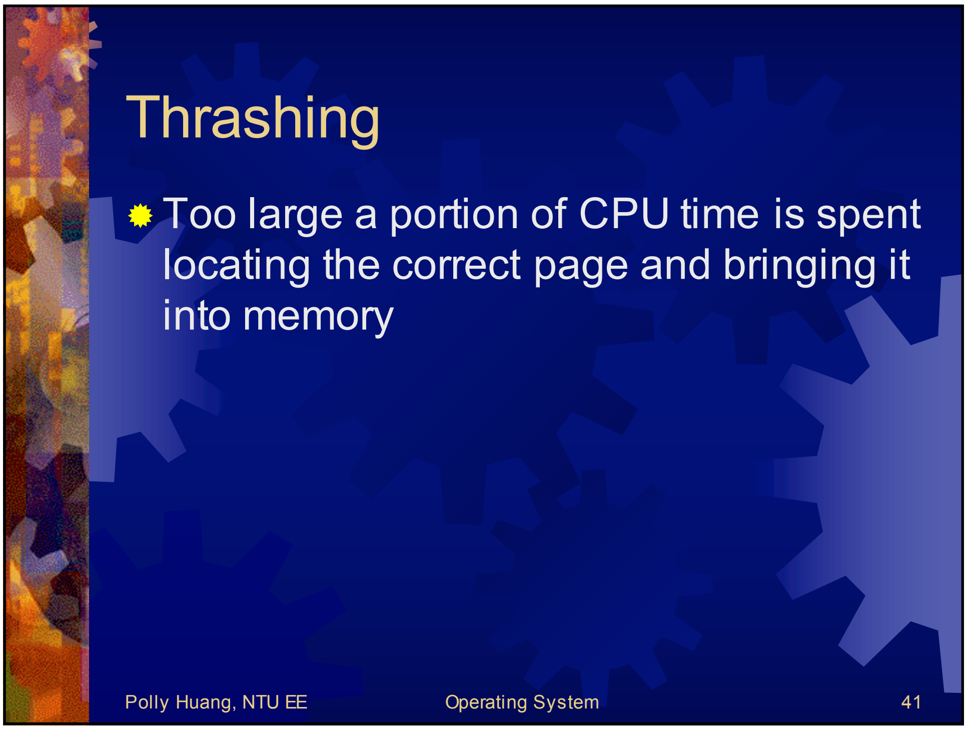
Swapping Rule

- Oldest
  - The oldest page gets swapped out first
- Least Frequently Used (LFU)
  - The least used gets swapped out first
- Least Recently Used (LRU)
  - The least recently used gets swapped out first
  - Linux



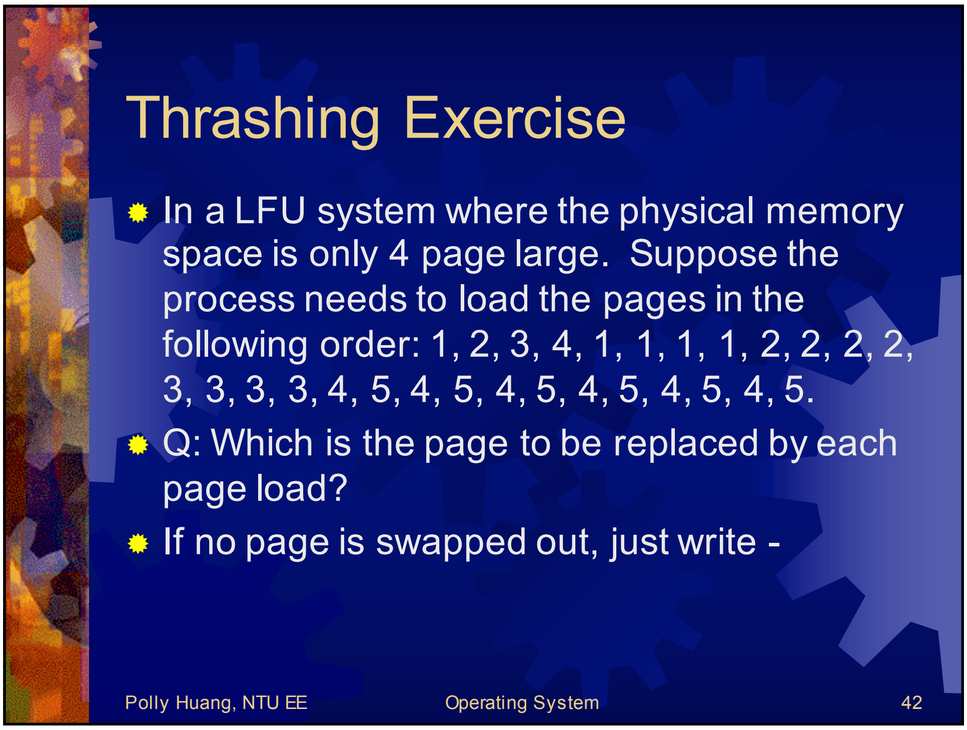
LRU Exercise

- There is a running process on Linux and the physical memory space is only 4 page large. Suppose the process needs to load the pages in the following order: 1, 2, 3, 4, 5, 2, 6.
  - Q: Which is the page to be replaced by page 5?
  - Q: Which is the page to be replaced by page 2?
  - Q: Which is the page to be replaced by page 6?



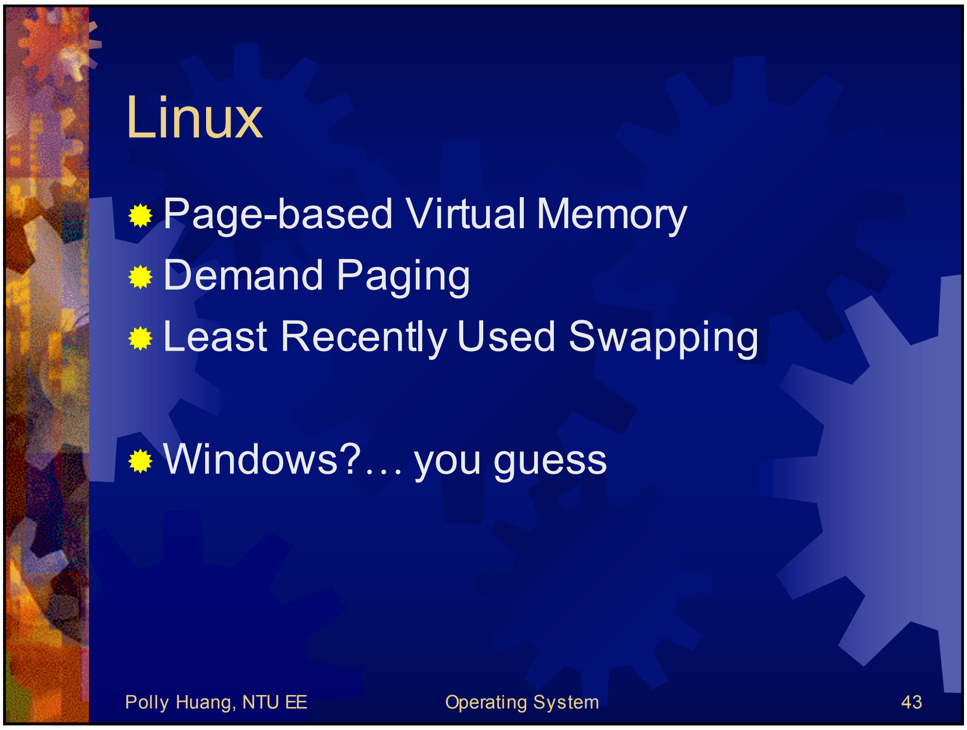
Thrashing

- Too large a portion of CPU time is spent locating the correct page and bringing it into memory



Thrashing Exercise

- In a LFU system where the physical memory space is only 4 page large. Suppose the process needs to load the pages in the following order: 1, 2, 3, 4, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5.
- Q: Which is the page to be replaced by each page load?
- If no page is swapped out, just write -



Linux

- Page-based Virtual Memory
- Demand Paging
- Least Recently Used Swapping

- Windows?… you guess

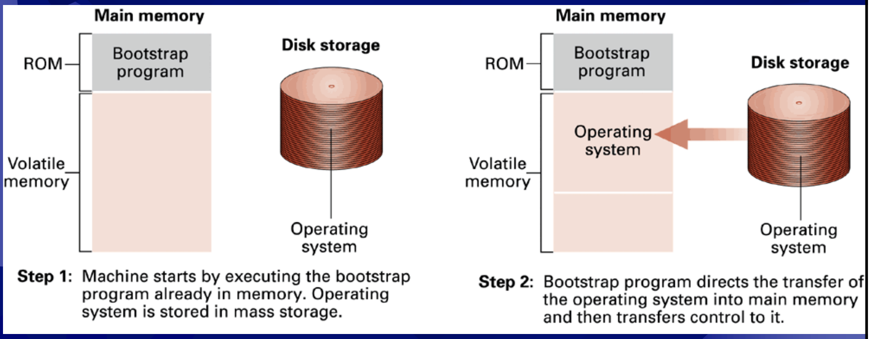
Scheduler and Dispatcher

- Process manager!
- Scheduler
  - Determines which activities are to be considered for execution in a time-sharing system
- Dispatcher
  - Controls the allocation of time slices to the scheduled activities (how long)

Getting it Started (bootstrapping)

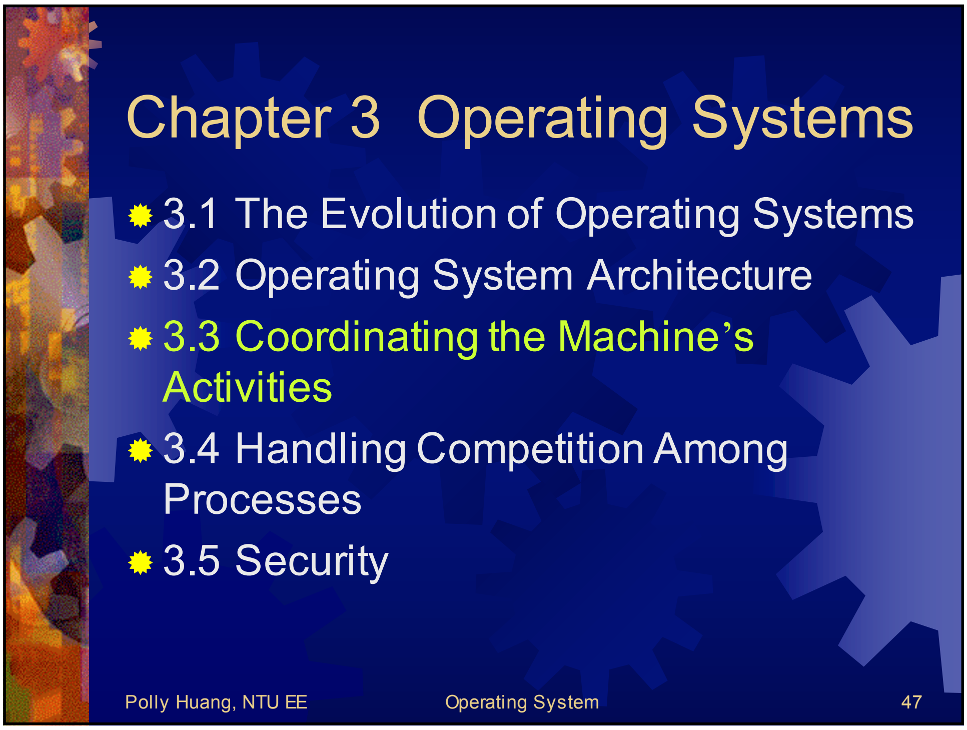
- Bootstrap: program in read only memory (ROM)
  - Run by the CPU when power is turned on
    - **B**asic **I**nput **O**utput **S**ystem
  - Transfers operating system from mass storage to main memory
    - After OS installation, inform the BIOS the location
    - Configurable via BIOS
  - Executes jump to operating system
  - At this point, the operating system takes over and begins controlling the machine’s behavior

The Booting Process



Step 1: Machine starts by executing the bootstrap program already in memory. Operating system is stored in mass storage.

Step 2: Bootstrap program directs the transfer of the operating system into main memory and then transfers control to it.



Chapter 3 Operating Systems

- 3.1 The Evolution of Operating Systems
- 3.2 Operating System Architecture
- 3.3 Coordinating the Machine's Activities
- 3.4 Handling Competition Among Processes
- 3.5 Security

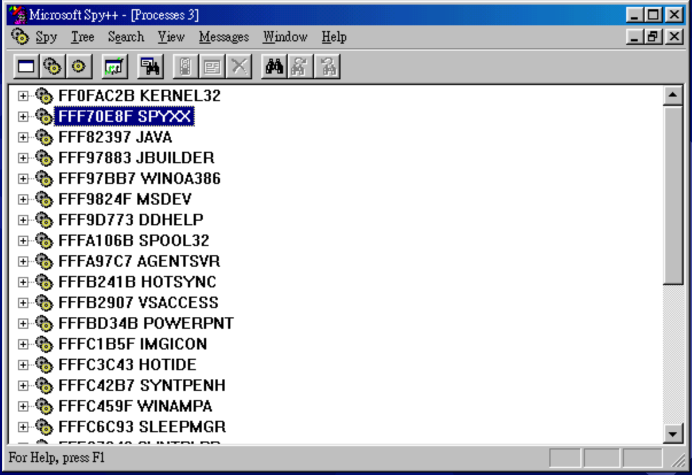
Processes

- Program
  - A static set of directions
- Process
  - Program (application and utilities) in execution
  - Needs resources to accomplish its task
  - Resources are given to the process when created
  - Can be executed in parallel

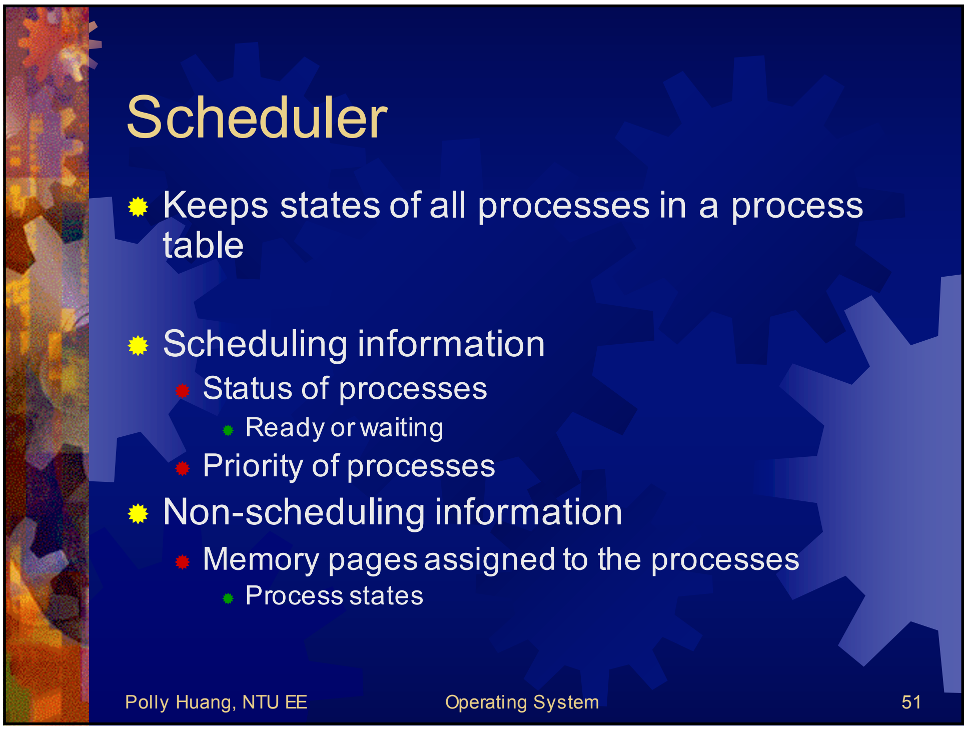
Process Management

- Handled by scheduler and dispatcher within kernel
- Scheduler
  - Maintains a process table
- Dispatcher
  - Ensures the scheduled processes are actually executed
  - Time slice (or quantum): typically no more than 50 milliseconds
  - Process switch

Process Table

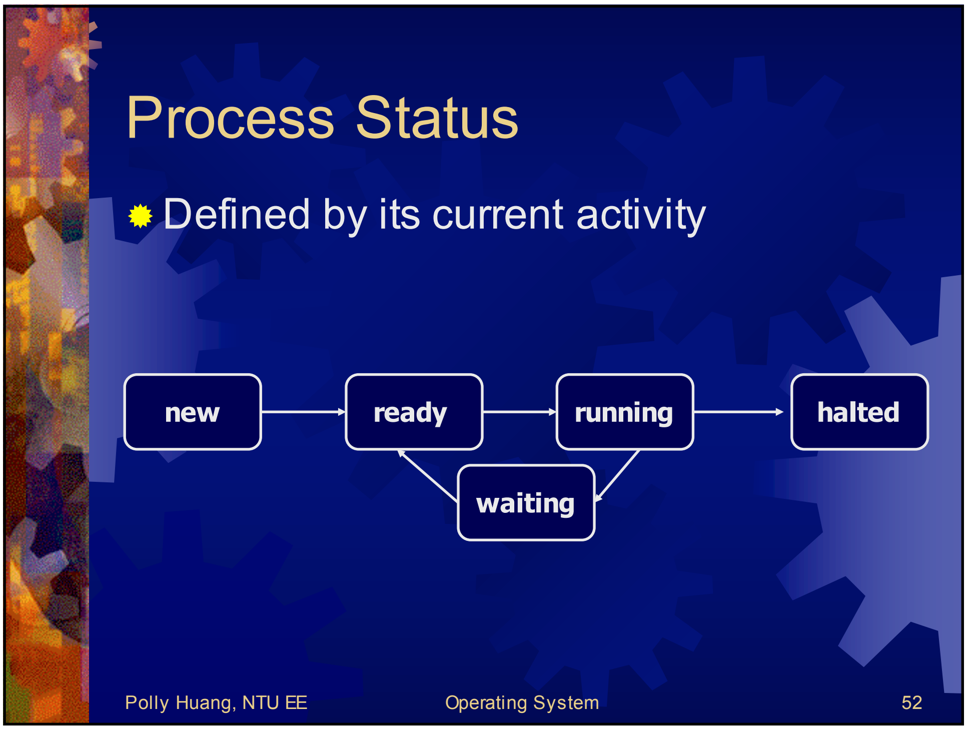

Microsoft Spy++ - [Processes 3]
Spy Tree Search View Messages Window Help
For Help, press F1

```
FF0FAC2B KERNEL32
FFF70E8F SPYXX
FFF82397 JAVA
FFF97883 JBUILDER
FFF97BB7 WINOA386
FFF9824F MSDEV
FFF9D773 DDHELP
FFFA106B SPOOL32
FFFA97C7 AGENTSVR
FFFB241B HOTSYNC
FFFB2907 VSACCESS
FFFBD34B POWERPNT
FFFC1B5F IMGICON
FFFC3C43 HOTIDE
FFFC42B7 SYNTPENH
FFFC459F WINAMPA
FFFC6C93 SLEEPMGR
```



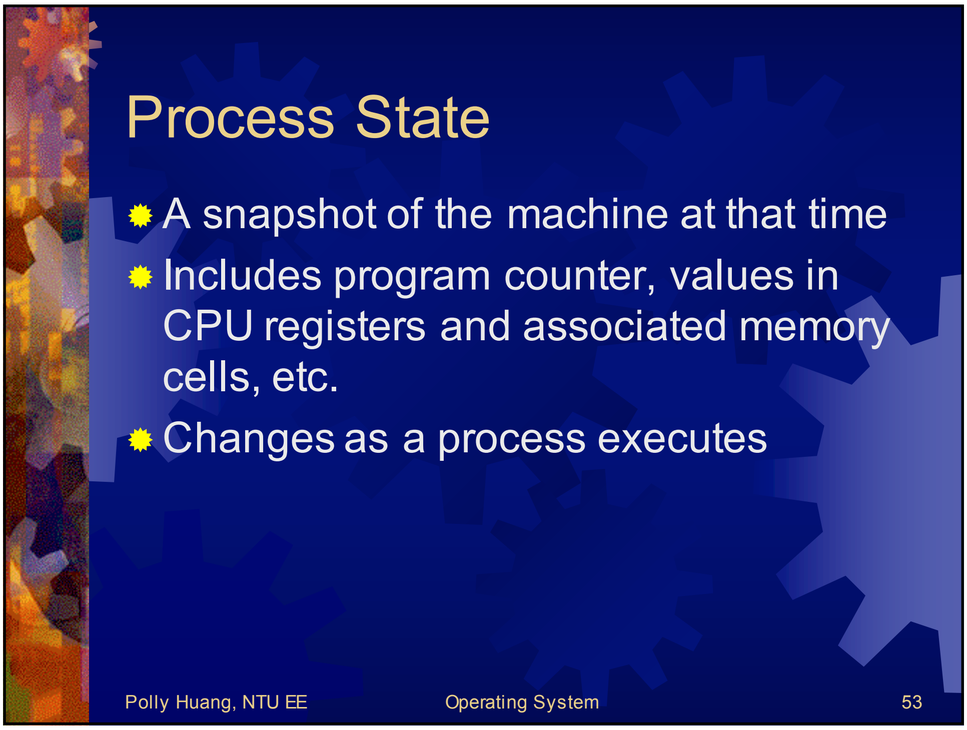
Scheduler

- Keeps states of all processes in a process table
- Scheduling information
  - Status of processes
    - Ready or waiting
  - Priority of processes
- Non-scheduling information
  - Memory pages assigned to the processes
    - Process states



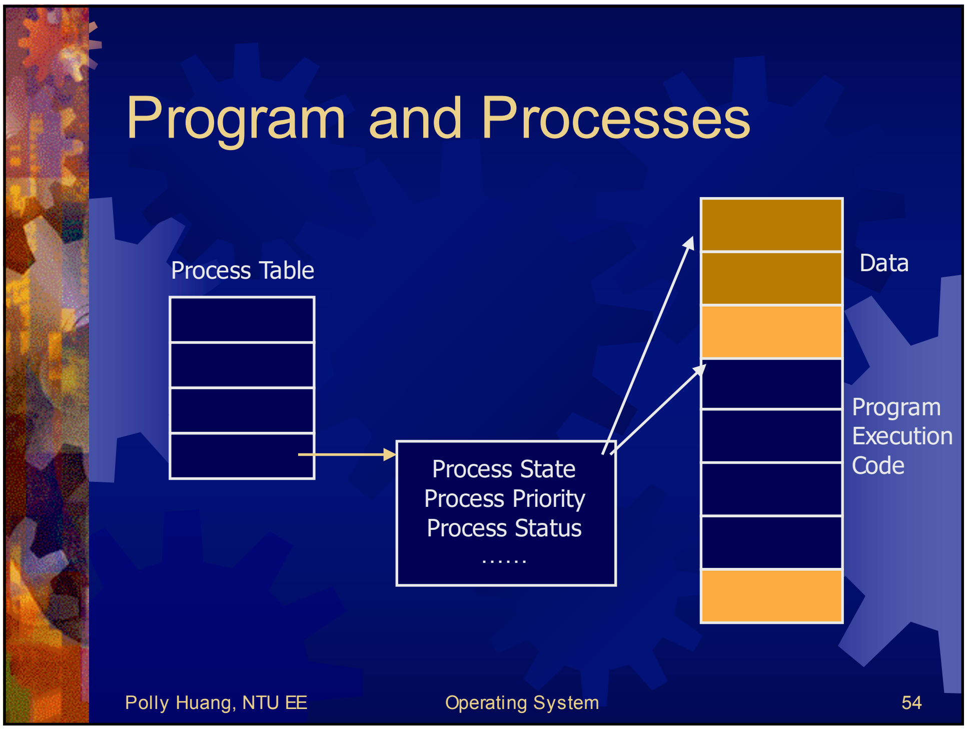
Process Status

- Defined by its current activity



Process State

- A snapshot of the machine at that time
- Includes program counter, values in CPU registers and associated memory cells, etc.
- Changes as a process executes



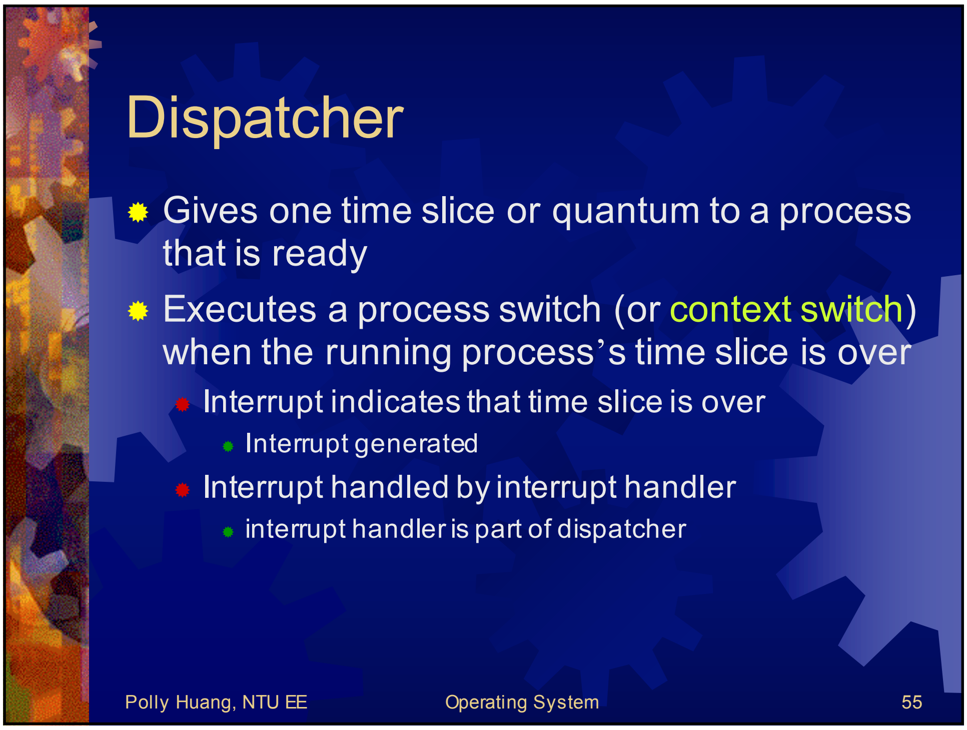
Program and Processes

Process Table

Process State
Process Priority
Process Status
……

Data

Program Execution Code

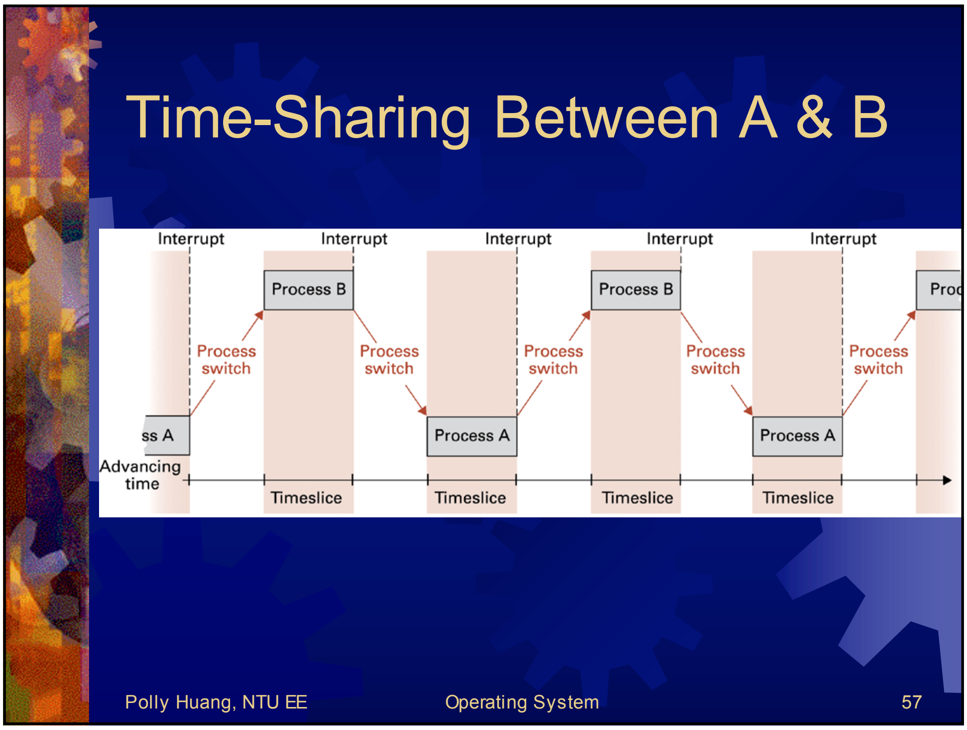


Dispatcher

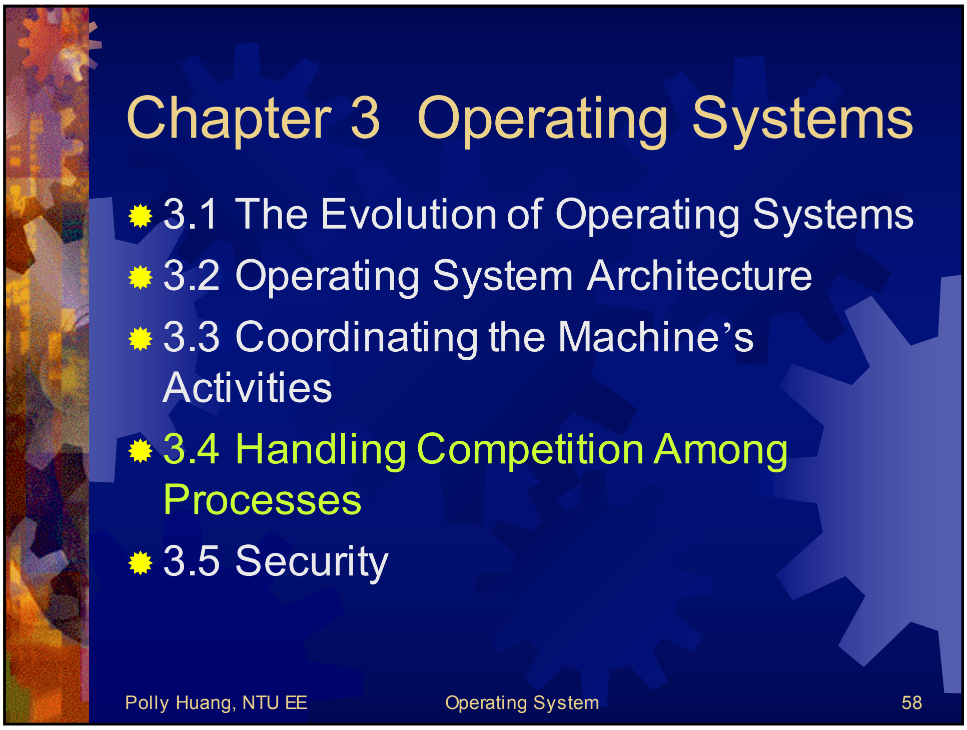
- Gives one time slice or quantum to a process that is ready
- Executes a process switch (or context switch) when the running process’s time slice is over
  - Interrupt indicates that time slice is over
    - Interrupt generated
  - Interrupt handled by interrupt handler
    - interrupt handler is part of dispatcher

Interrupt Handler

- Process state of an executing process is saved and the process becomes idle
- Process state of the process to be executed next is loaded
- The new process starts running

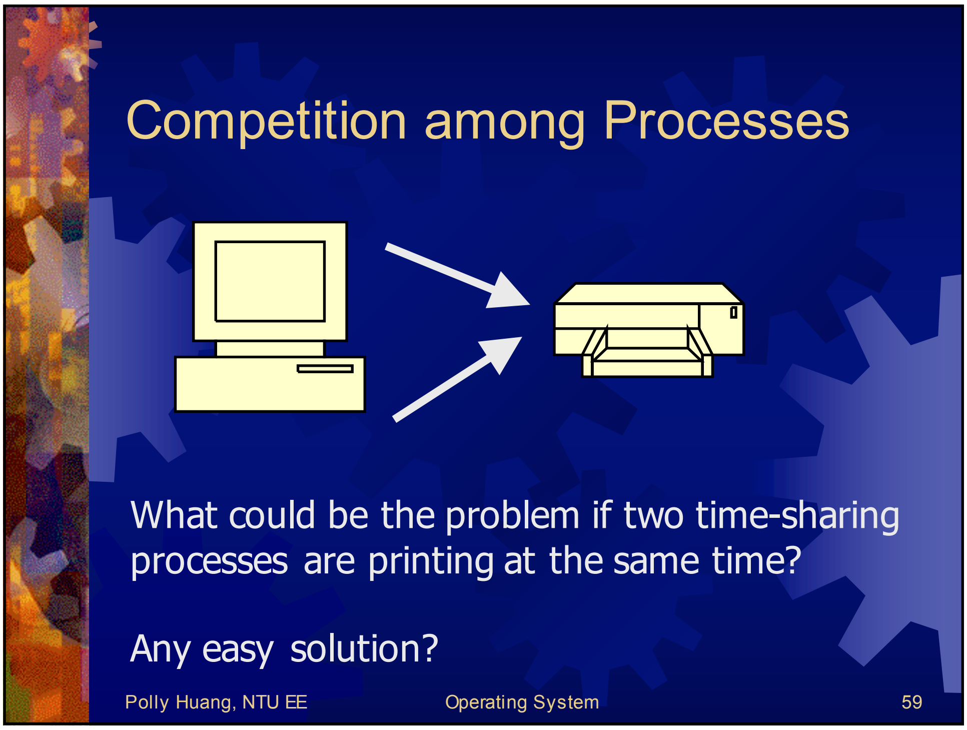


Time-Sharing Between A & B



Chapter 3 Operating Systems

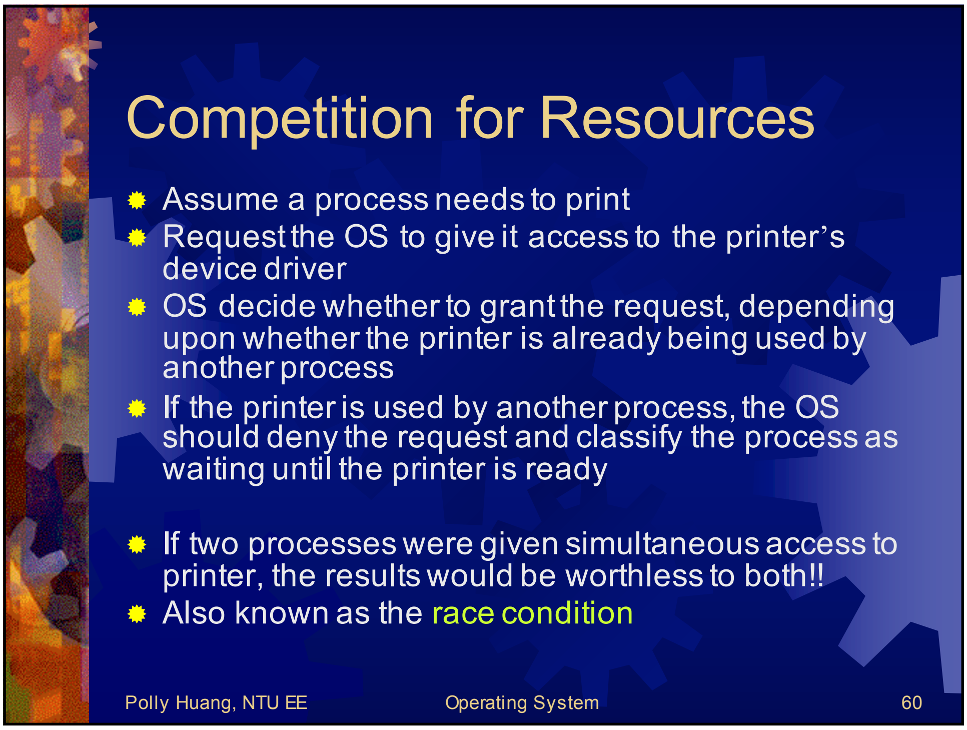
- 3.1 The Evolution of Operating Systems
- 3.2 Operating System Architecture
- 3.3 Coordinating the Machine's Activities
- 3.4 Handling Competition Among Processes
- 3.5 Security



Competition among Processes

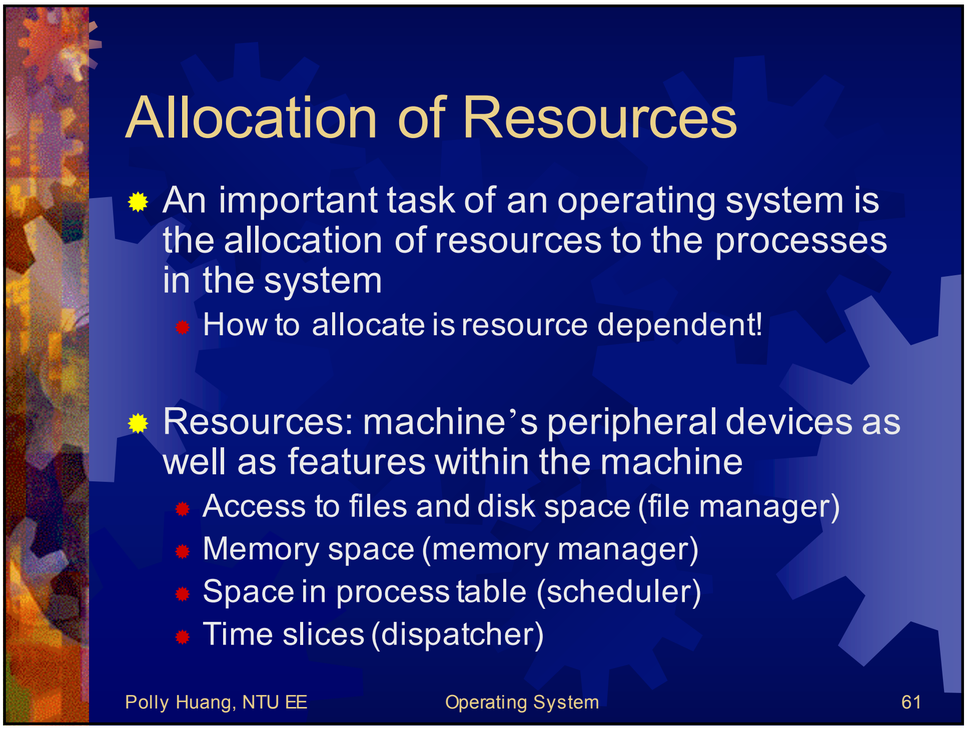
What could be the problem if two time-sharing processes are printing at the same time?

Any easy solution?



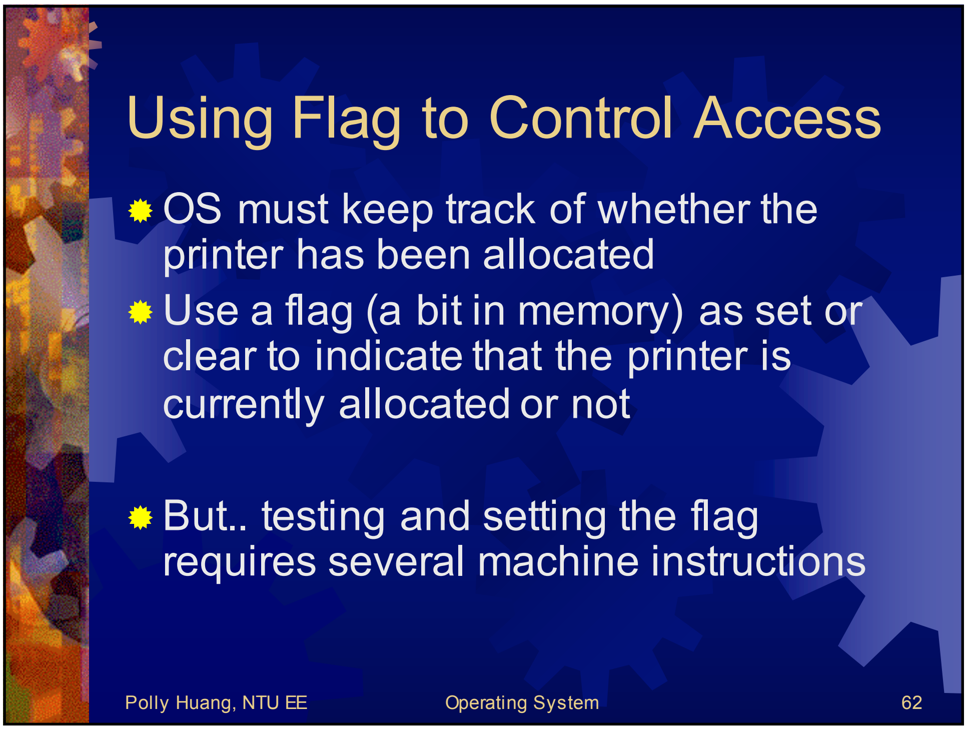
Competition for Resources

- Assume a process needs to print
- Request the OS to give it access to the printer's device driver
- OS decide whether to grant the request, depending upon whether the printer is already being used by another process
- If the printer is used by another process, the OS should deny the request and classify the process as waiting until the printer is ready
- If two processes were given simultaneous access to printer, the results would be worthless to both!!
- Also known as the race condition



Allocation of Resources

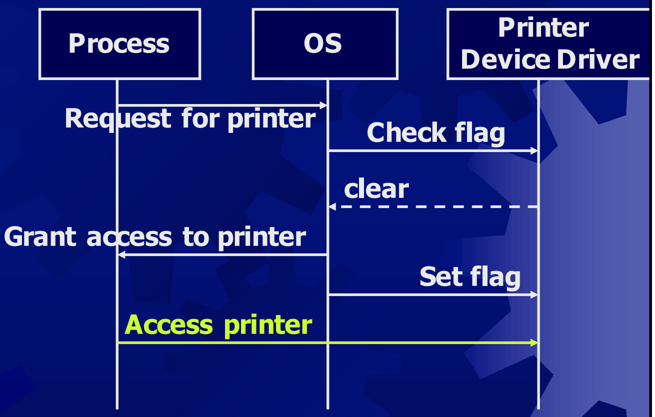
- An important task of an operating system is the allocation of resources to the processes in the system
  - How to allocate is resource dependent!
- Resources: machine’s peripheral devices as well as features within the machine
  - Access to files and disk space (file manager)
  - Memory space (memory manager)
  - Space in process table (scheduler)
  - Time slices (dispatcher)



Using Flag to Control Access

- OS must keep track of whether the printer has been allocated
- Use a flag (a bit in memory) as set or clear to indicate that the printer is currently allocated or not
- But.. testing and setting the flag requires several machine instructions

Flag Setting: Step by Step



Process

OS

Printer Device Driver

Request for printer

Check flag

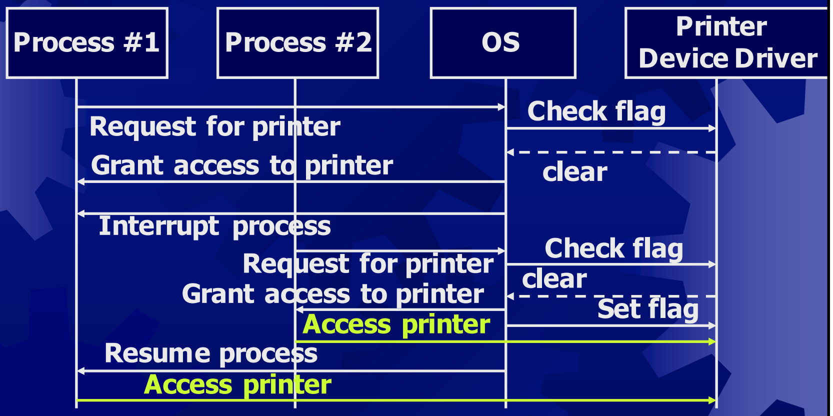
clear

Grant access to printer

Set flag

Access printer

A Possible Problem



Process #1

Process #2

OS

Printer Device Driver

Request for printer

Check flag

Grant access to printer

clear

Interrupt process

Check flag

Request for printer

clear

Grant access to printer

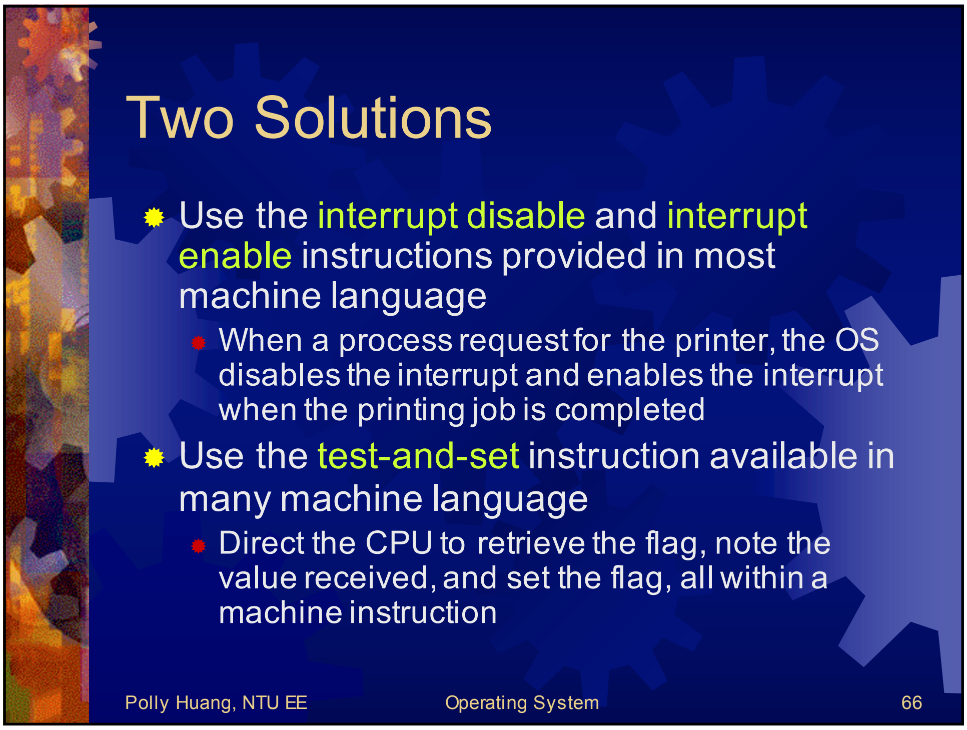
Set flag

Access printer

Resume process

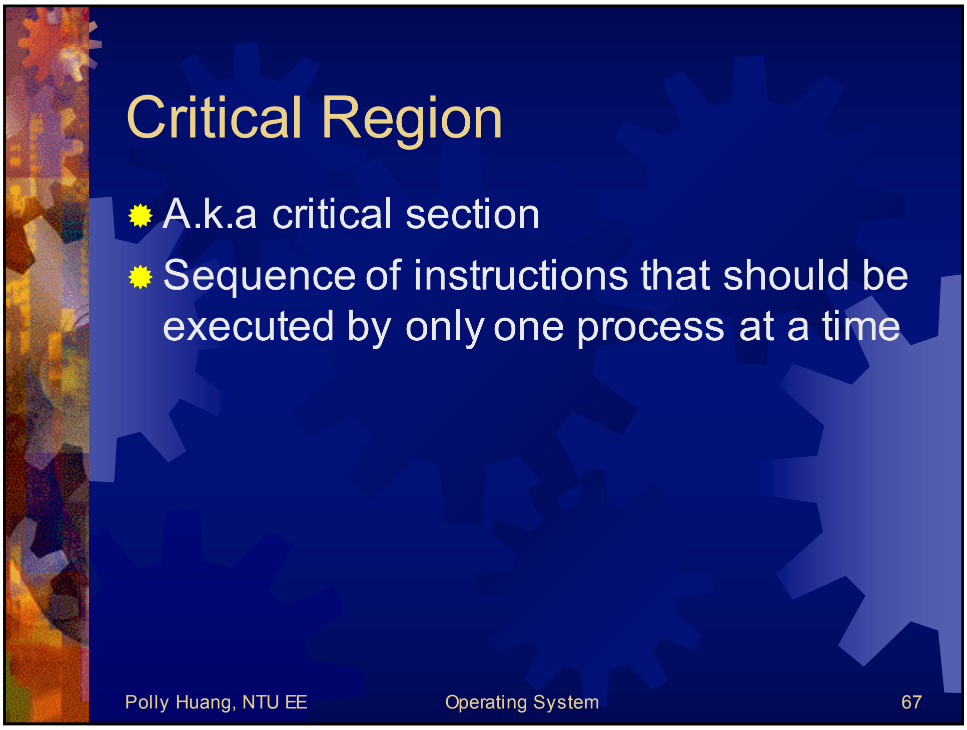
Access printer

Quiz Time!



Two Solutions

- Use the interrupt disable and interrupt enable instructions provided in most machine language
  - When a process request for the printer, the OS disables the interrupt and enables the interrupt when the printing job is completed
- Use the test-and-set instruction available in many machine language
  - Direct the CPU to retrieve the flag, note the value received, and set the flag, all within a machine instruction



Critical Region

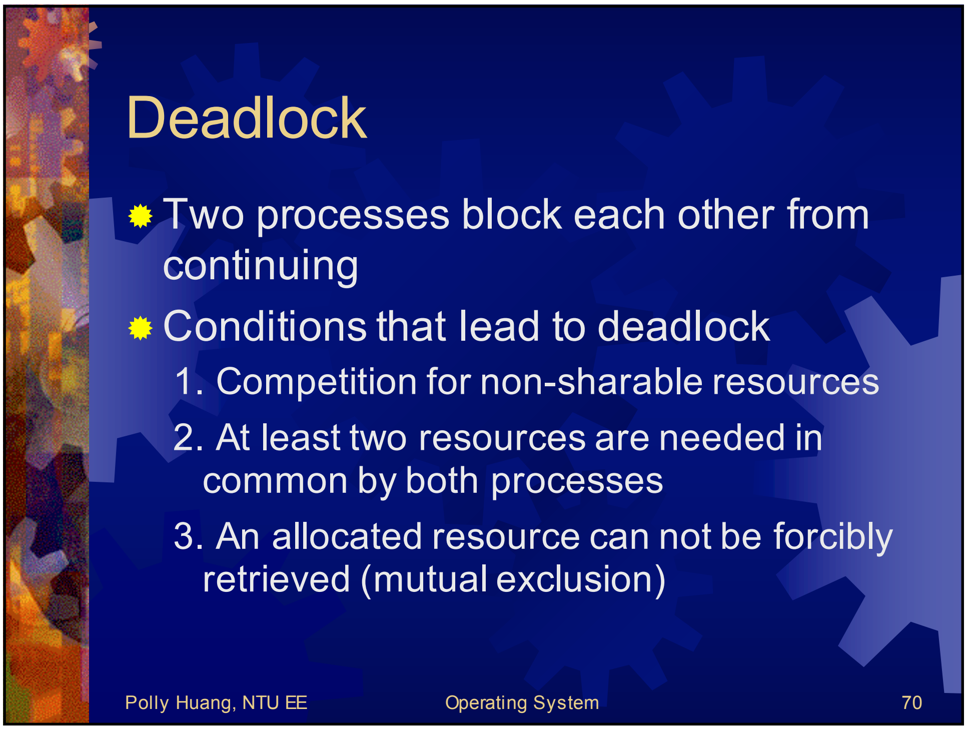
- A.k.a critical section
- Sequence of instructions that should be executed by only one process at a time

Semaphore

- A control flag telling if resource is in use
- Test and set must be done together for non-preemptable resources (e.g., printer)

Mutual Exclusion

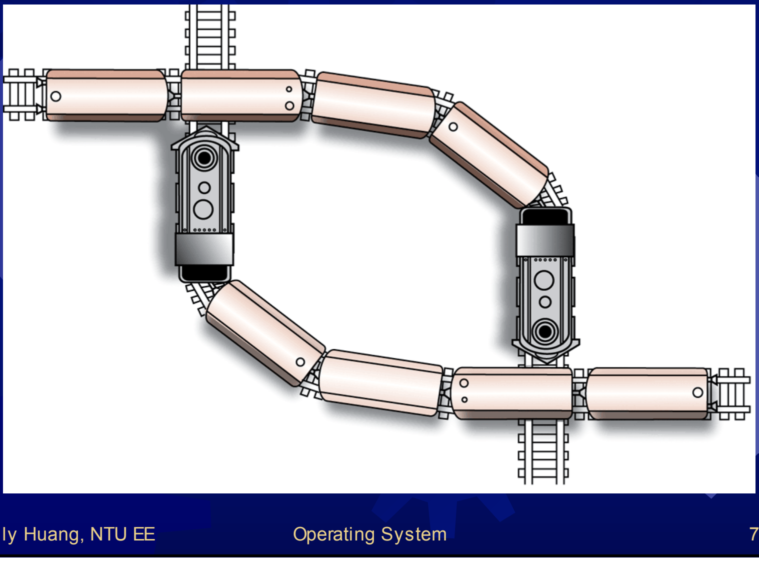
- One process already in the critical region
- Other processes cannot enter the critical region
- But…



Deadlock

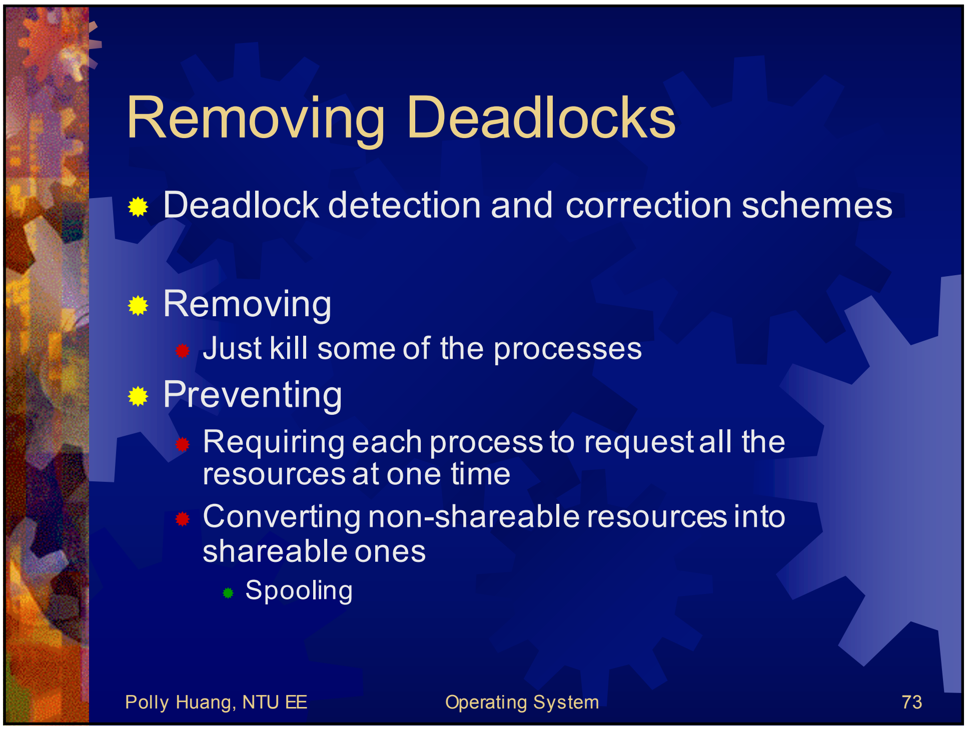
- Two processes block each other from continuing
- Conditions that lead to deadlock
  1. Competition for non-sharable resources
  2. At least two resources are needed in common by both processes
  3. An allocated resource can not be forcibly retrieved (mutual exclusion)

A Deadlock



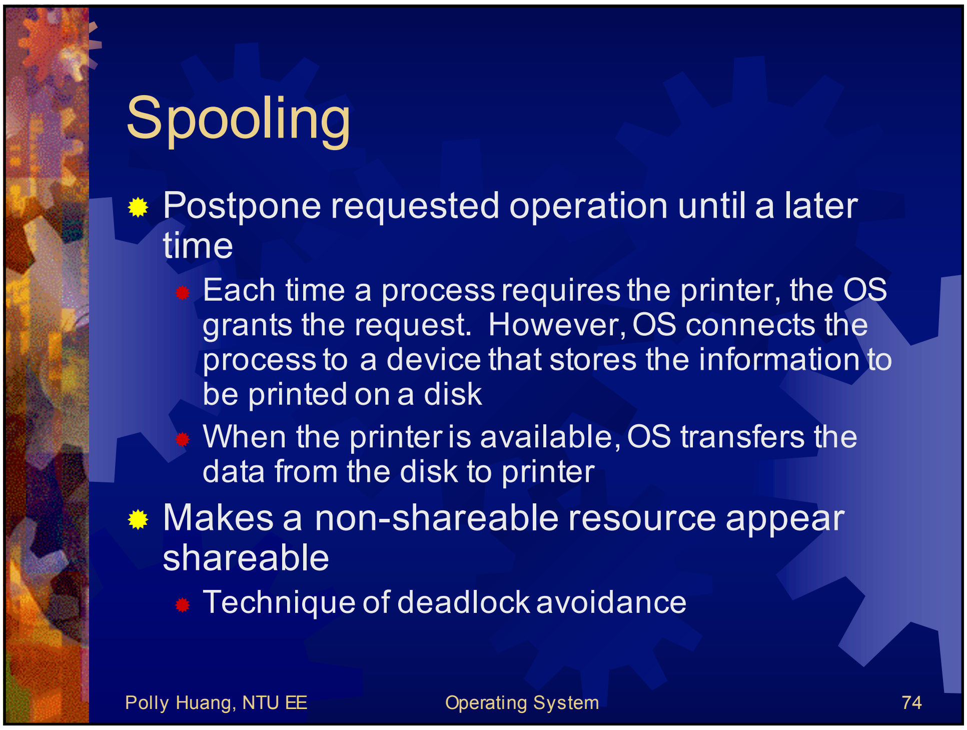
Examples of Deadlock

- One process has access to printer but is waiting for tape drive, while another process has access to tape drive but is waiting for printer
- Scheduler has no space left in the process table and each process in the system must create an additional process before it can complete its task
- Other examples of deadlock in real life?



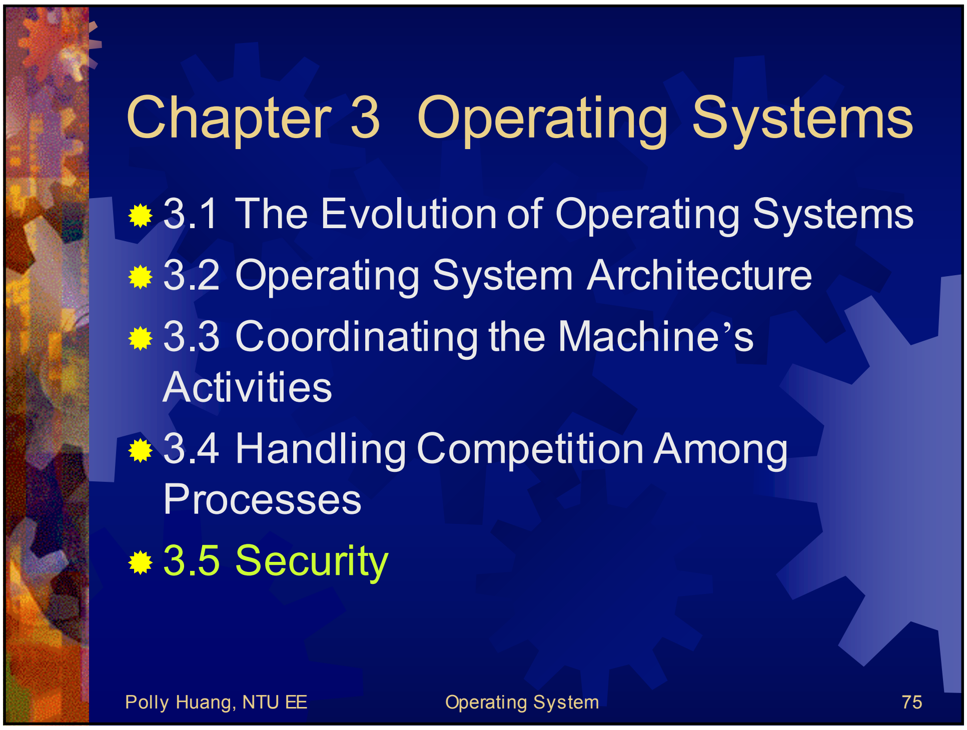
Removing Deadlocks

- Deadlock detection and correction schemes
- Removing
  - Just kill some of the processes
- Preventing
  - Requiring each process to request all the resources at one time
  - Converting non-shareable resources into shareable ones
    - Spooling



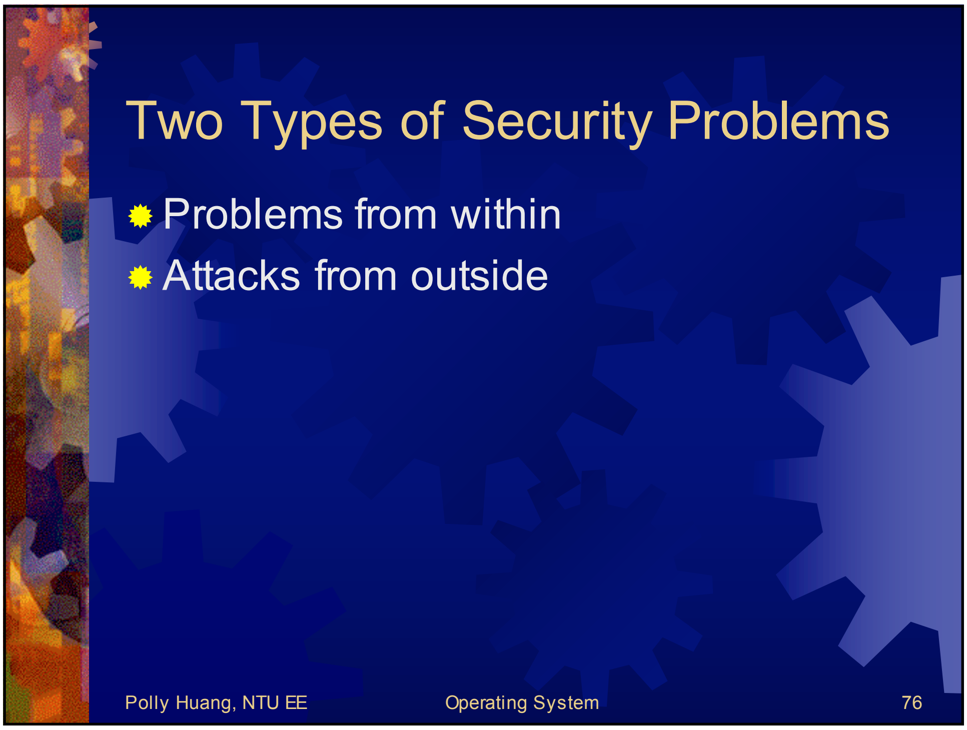
Spooling

- Postpone requested operation until a later time
  - Each time a process requires the printer, the OS grants the request. However, OS connects the process to a device that stores the information to be printed on a disk
  - When the printer is available, OS transfers the data from the disk to printer
- Makes a non-shareable resource appear shareable
  - Technique of deadlock avoidance



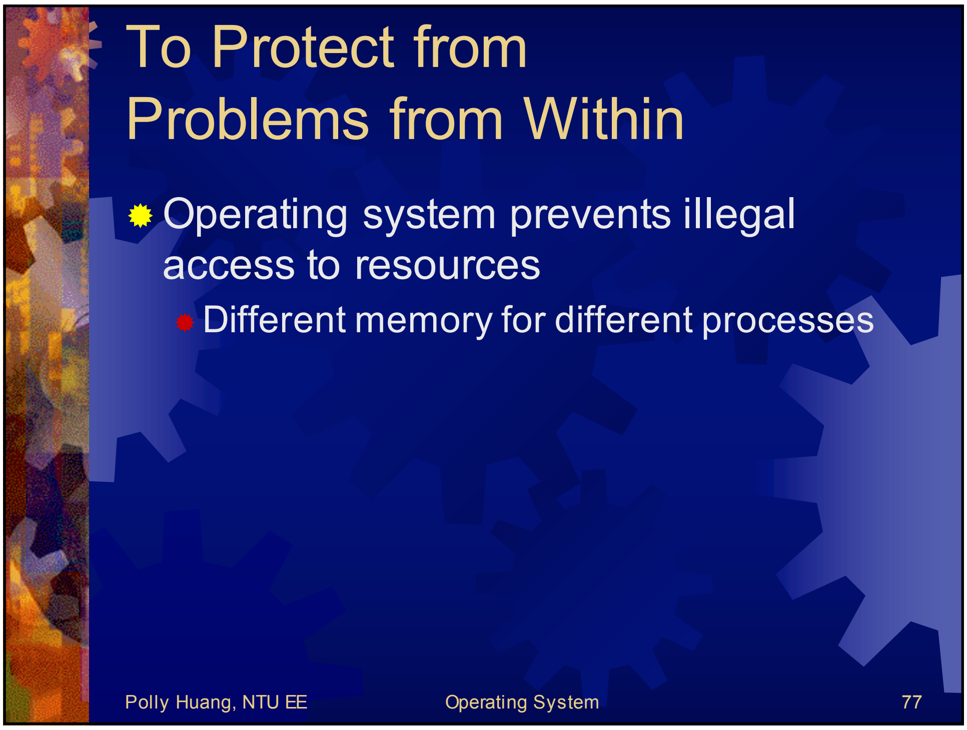
Chapter 3 Operating Systems

- 3.1 The Evolution of Operating Systems
- 3.2 Operating System Architecture
- 3.3 Coordinating the Machine's Activities
- 3.4 Handling Competition Among Processes
- 3.5 Security



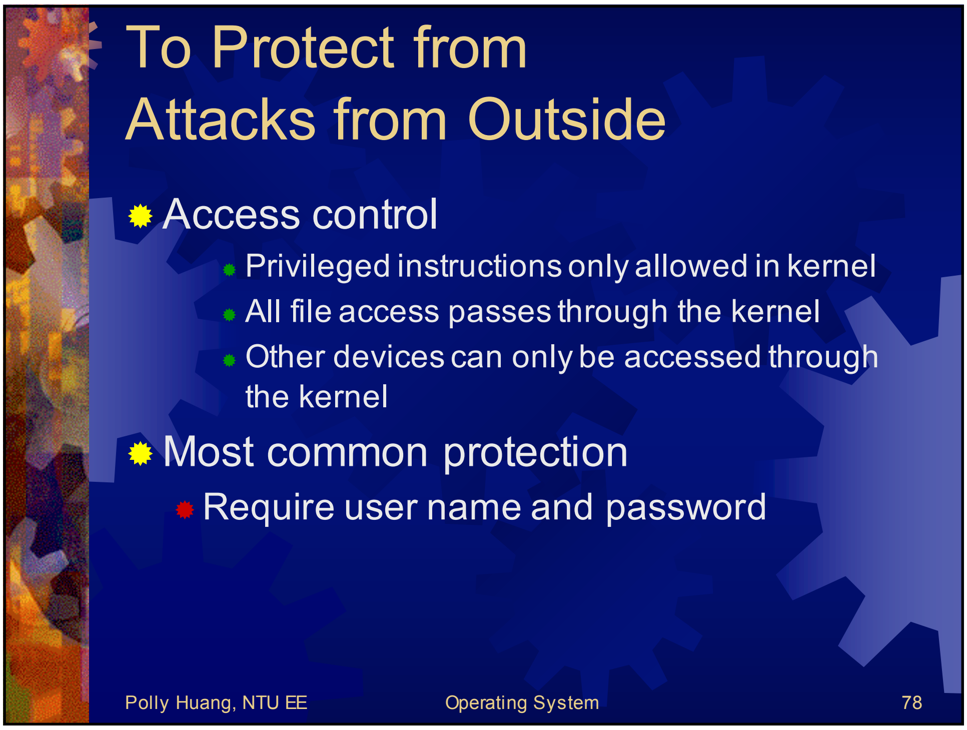
Two Types of Security Problems

- Problems from within
- Attacks from outside



To Protect from Problems from Within

- Operating system prevents illegal access to resources
  - Different memory for different processes

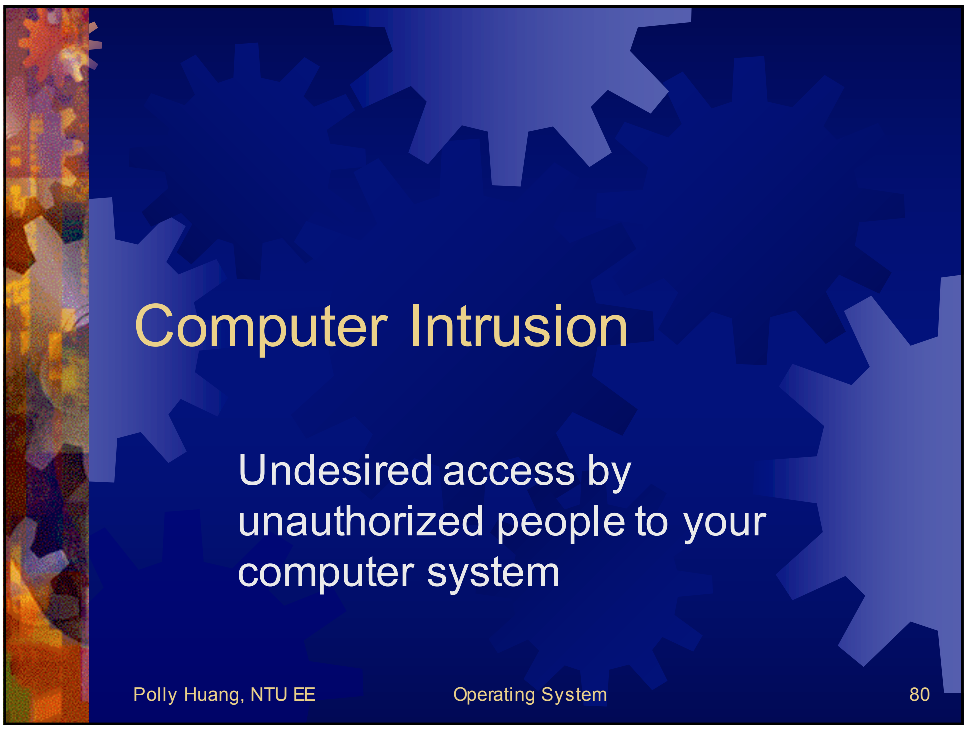


To Protect from Attacks from Outside

- Access control
    - Privileged instructions only allowed in kernel
    - All file access passes through the kernel
    - Other devices can only be accessed through the kernel
- Most common protection
  - Require user name and password

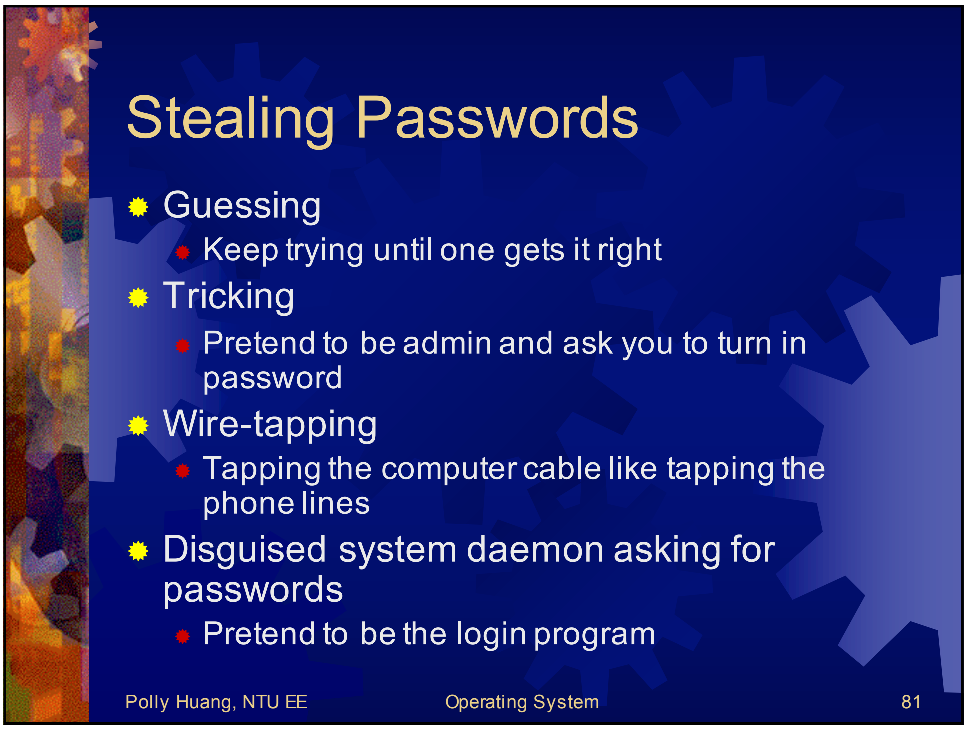
Hot Attacks

- Intrusion
- The pests



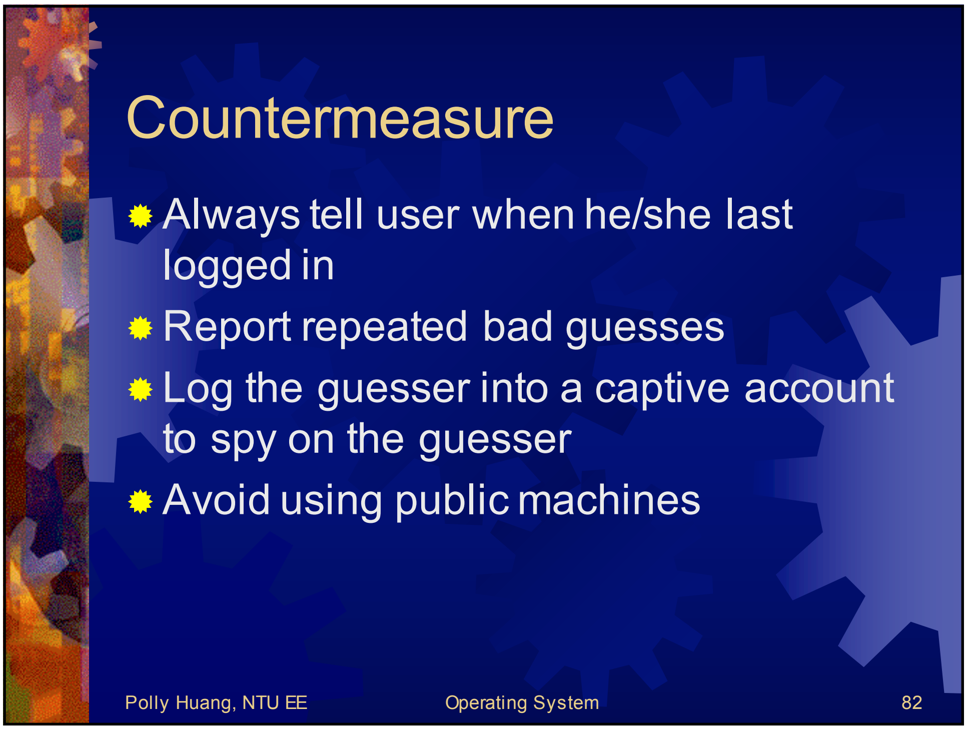
Computer Intrusion

Undesired access by unauthorized people to your computer system



Stealing Passwords

- Guessing
  - Keep trying until one gets it right
- Tricking
  - Pretend to be admin and ask you to turn in password
- Wire-tapping
  - Tapping the computer cable like tapping the phone lines
- Disguised system daemon asking for passwords
  - Pretend to be the login program



Countermeasure

- Always tell user when he/she last logged in
- Report repeated bad guesses
- Log the guesser into a captive account to spy on the guesser
- Avoid using public machines

Questions?

Quiz Time!