Instructions for use: These knowledge points are only I personally feel that the possibility of examination is relatively high, and there is no official basis, that is to say, these may not be test sites. Please use with caution!

Chapter 1&2

1.What is OS?

Operating System is the System Software to control and management the software and hardware's resource of computer. It also is the interface between User and Computer.

2.User mode and Kernel mode.

In order to make sure that the error of user process only influent the running program, it has much necessary to make difference between OS code and User code.

3.What is system call?

System Call is a special interface. User can access the kernel space via system call.

Progress: User process ->System call->Kernel->return User space

4.Computer System consists of 4 parts: Hardware, OS, System program and Application, User

Chapter 3 Process

1.What is process?

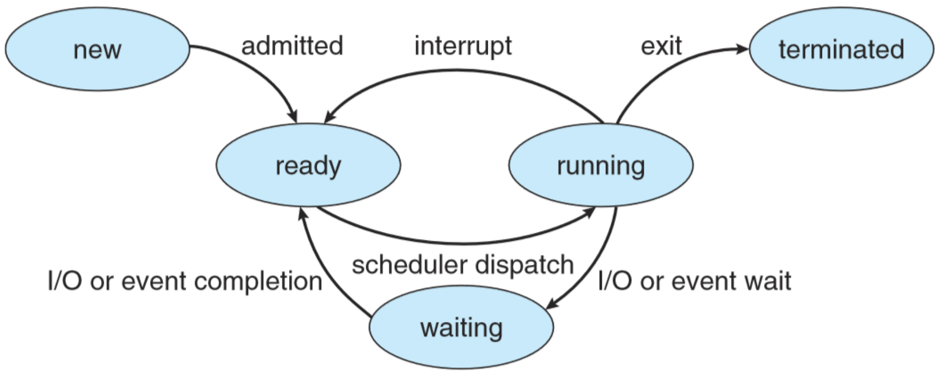
Running program.

2.The difference between Process and Program.

Process: dynamic   Program: Static

Only the .exe file is put into the memory, can it be called Process.

3.States of Process



4.PCB(Process Control Block)

5.Process schedule

Select a useful process to be executed on CPU.

6.Job queue & Ready queue

Job queue: the process that in the system.

Ready queue: in the memory, wait to be run.

7.Scheduling

Long-term scheduling(Job scheduling): select a process go into the memory. Not frequently.

Short-term(CPU scheduling): select a process from the Ready queue and allocate CPU for it. Frequently.

8.Context switch

When CPU switch another process, system should save old process's state and load new process.

9.Process classification: I/O-based processes, CPU-based processes.

10.Terminate a process

1.Search its PCB from the PCB set and read out its state.

2.If it is running, terminate it and select a new process to use the CPU.

3.Terminate its child-processes. In case they become the orphan process.

4.Release all the resource and PCB.

5.Remove it from the queue where its PCB in.

11. Basic mode of interprocess communication: (1) shared memory (2) messaging

Chapter 4 Thread

1.What is thread?

It is the basic unit of CPU running. It is an entity in the process, is the basic unit of independent scheduling and dispatch.

2.Thread can share the code segment, data segment, operating system resource with the threads belonging the same process.

3.The difference between User thread and Kernel thread.(3 points)

4.Three models.

Many to one.

One to one.

Many to many.

Two-Level-model.

5.Parallelism and Concurrent.

Chapter 5

1.When CPU schedule happens?

I/O event, I/O event completion, interrupt, exit

2.Scheduling algorithms

FCFS, SJF, SRT, priority scheduling, Round-Robin Scheduling, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling and they're difference.

Chapter 6 Process Synchronization

1.Critical-Section Problem

2.Three conditions to solve the Critical-Section Problem.

1.Mutual exclusion

2.Progress

3.Bounded waiting

3.Semaphores

4.Deadlocks and Starvation

5.The Bounded-Buffer Problem

The Dining-Philosophers Problem and they’re solution

Chapter 7 Deadlocks

1.Necessary Conditions

1.Mutual exclusion

2.Hold and wait

3.No preemption

4.Circular waiting

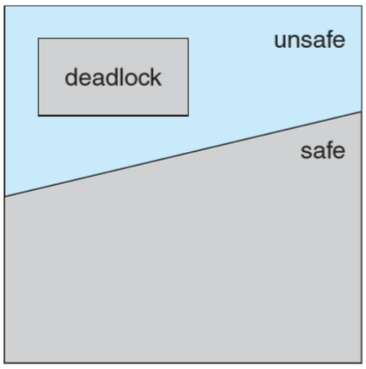
2.Resource-Allocation Graph

3.Deadlock Prevention

Three ways

4.Deadlock Avoidance

Safe State



6.Banker’s Algorithm

Is a deadlock prevention algorithm

Chapter 8 Memory management

1.Logical address and physical address

2.Address Binding

3.Dynamic Loading

4.Static Linking

5.Dynamic Linking

6.Contiguous memory allocation

First fit, Best fit, Worst fit

7.Fragmentation

8.Segmentation

9.Paging

TLB

10.The way to accomplish the address relocation

1.Static address relocation

2.Dynamic address relocation

Chapter 9 Visual Memory

1. What’s Visual memory and why use it?

2.Demanding Paging

3.What can we do if a page fault happened?

Computation:

effective access time = (1−p)×(200) + p (8 milliseconds) = (1−p)×200 + p×8,000,000 = 200 + 7,999,800×p.

Steps :

1. We check an internal table (usually kept with the process control block) for this process to determine whether the reference was a valid or an invalid memory access.

2. If the reference was invalid, we terminate the process. If it was valid but we have not yet brought in that page, we now page it in.

3. We find a free frame (by taking one from the free-frame list, for example).

4. We schedule a disk operation to read the desired page into the newly allocated frame.

5. When the disk read is complete, we modify the internal table kept with the process and the page table to indicate that the page is now in memory.

6. We restart the instruction that was interrupted by the trap. The process can now access the page as though it had always been in memory.

4.Page Replacement algorithms

FIFO, OPR, LRU

5.Frame Allocation Algorithms

6.Thrashing

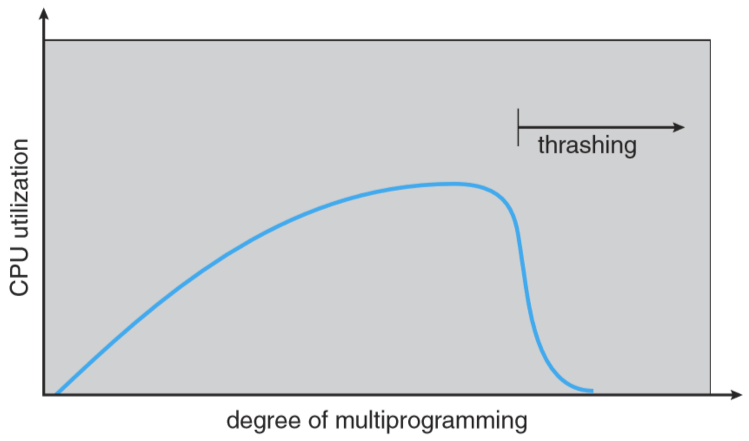
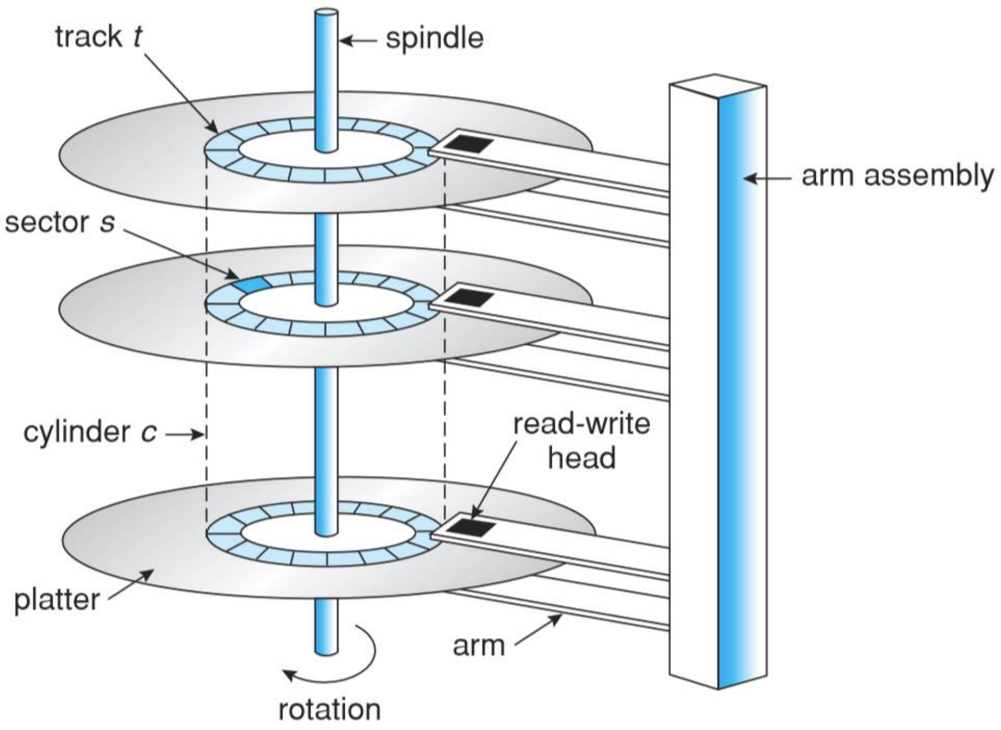


Figure 9.18 Thrashing.

Chapter 10 Mass-Storage Structure

1.Disk Structure



2.Disk Scheduling

FCFS, SSTF, SCAN, C-SCAN, LOOK

By 1702 Zerlin Wang