名词解释提纲：

1. Logical and physical address
   1. **Logical address** – generated by the CPU; also referred to as **virtual address**
   2. **Physical address** – address seen by the memory unit
2. System Call

User request for operating system service

1. File

Abstract from physical properties of storage devices to define a logical storage unit – file

1. Safe State

the system can allocate resources to each process (up to its maximum) in some order and still avoid a deadlock

1. Critical Section

A part of program where resources are shared by other process

1. Deadlock

A set of blocked processes each holding a resource and waiting to acquire a resource held by another blocked process in the set

1. Semaphore

Synchronization tool for processes to synchronize their activities.

1. External fragmentation

As processes are loaded and removed from memory, the free memory space is broken into little pieces. Total memory space exists to satisfy a request, but it is not contiguous, the storage is fragmented into a large number of small holes

1. Program and Process

A **program** that acts as an intermediary between the computer user and the computer hardware

A **process** is an execution of a program;

10、Virtual memory

Virtual memory – separation of user logical memory from physical memory.

Only part of the program needs to be in memory for execution

11、Context switch

When CPU switches to another process, the system must **save the state** of the old process and perform a **state restore** for the new process via a **context switch**

12、Demand paging

Could load entire process into physical memory at execution time

May not initially need the entire program in memory

Or load pages into memory only when they are needed

简答题提纲：

1. What is paging? What is segmentation? What are the main differences between the two?

**Paging** is a memory-management scheme that permits the physical address space of a process to be noncontiguous. Basic method for implementing paging involves breaking physical memory into frames and breaking logical memory into pages. Each address are consisted of two quantities: a page number and an offset.

**Segmentation** is a memory-management scheme that supports this user view of memory. A logical address space is a collection of segments. Each segment has a name and a length. The addresses specify both the segment name and the offset within the segment. The user therefore specifies each address by two quantities: a segment name and an offset.

1. What are three major methods of allocating disk space in file-system? What are their characteristics?

Contiguous, linked, and indexed.

(1) Contiguous allocation

Contiguous allocation requires that each file occupy a set of contiguous blocks on the disk. It is simple because it is only defined by starting location of block and length (number of blocks). Accessing a file that has been allocated contiguously is easy. For sequential access, the file system remembers the disk address of the last block referenced and, when necessary, reads the next block. Contiguous allocation has some problems, however. One difficulty is finding space for a new file. Second is dynamic storage-allocation problem. Third problem is determining how much space is needed for a file. If we allocate too little space to a file, we may find that the file cannot be extended.

(2) Linked allocation

With linked allocation, each file is a linked list of disk blocks; the disk blocks may be scattered anywhere on the disk. The directory contains a pointer to the first and last blocks of the file. There is no external fragmentation with linked allocation, and any free block on the free-space list can be used to satisfy a request. Linked allocation does have disadvantages, however. The major problem is that it can be used effectively only for sequential-access files. Another disadvantage is the space required for the pointers.

(3) Indexed allocation

Indexed allocation solves this problem by bringing all the pointers together into one location: the index block. Indexed allocation supports direct access, without suffering from external fragmentation. Indexed allocation does suffer from wasted space, however. The pointer overhead of the index block is generally greater than the pointer overhead of linked allocation.

1. What are two-address-space possibilities for the new process created by fork()?
2. In the RR scheduling algorithm, how does the time slice value (quantum) of the system influence on the system’ performance?

The value of time slice is related to the efficiency of computer system and user satisfaction, so the value of time slice should be determined according to the response time required by the process and the number of processes entering the system.

If the system is required to respond quickly, the time slice is smaller, which reduces the turnaround time and allows the process to respond as quickly as possible. If the number of processes is small, the time slice can be larger, which can reduce the times of process scheduling and improve the efficiency of the system.

However, it should be noted that when the time slice value is too large, the algorithm degenerates into a first-come-first-served algorithm. If the time slice value is too small, it will increase system overhead. Generally, 80% of CPU bursts should be shorter than time quantum

1. What are three structures of the page table? Describe their characteristics.

**Hierarchical Paging：**Break up the logical address into multiple page tables.Page table itself is also page

**Hashed Page Tables：**The virtual page number is hashed into a page table

This page table contains a linked list of elements that hashing to the same location

**Inverted Page Tables：**Rather than each process having a page table and keeping track of all possible logical pages, track all physical pages

One entry for each real page of memory

Entry consists of the virtual address of the page stored in that real memory location, with information about the process that owns that page

1. Describe the basic steps in handing a page fault in demand paging.
2. What is deadlock? Please illustrate the cause of deadlock and the necessary conditions for deadlock.

A set of blocked processes each holding a resource and waiting to acquire a resource held by another blocked process in the set

**Mutual exclusion:** only one process at a time can use a resource

**Hold and wait:** a process holding at least one resource is waiting to acquire additional resources held by other processes

**No preemption:** a resource can be released only voluntarily by the process holding it, after that process has completed its task

**Circular wait:** there exists a set {*P*0, *P*1, …, *P*n} of waiting processes such that *P*0 is waiting for a resource that is held by *P*1, *P*1 is waiting for a resource that is held by *P*2, …, *Pn*–1 is waiting for a resource that is held by *P*n, and *P*n is waiting for a resource that is held by *P*0.

1. How does a TLB (translation-lookaside buffer) speed the process of address translation?

The two memory access problem can be solved by the use of a special fast-lookup hardware cache called associative memory or translation look-aside buffers

1. Describe the actions taken by a kernel to context-switch between processes.

**Save Current State:** Save the state of the current process (registers, program counter) in its Process Control Block (PCB).

**Update Process State:** Change the current process state to "waiting" or "ready."

**Select Next Process:** Choose the next process to run from the ready queue.

**Load Memory Management Info:** Load the memory management data (page table) for the next process.

**Restore Next Process State:** Restore the state of the next process from its PCB.

Switch to User Mode: Switch the CPU to user mode and start executing the next process.

1. What mechanism does an operating system provides to build the correlation between API and associated system call within the kernel? How it works?

The run-time support system (a set of functions built into libraries included with a compiler) provides a system-call interface that serves as link to system calls made available by the operating system

The system-call interface intercepts function calls in the API and invokes the necessary system call within the operating system

Typically, a number is associated with each system call, and the system-call interface maintains a table indexed according to these numbers

The system call interface then invokes intended system call in OS kernel and returns status of the system call and any return values

1. What are the typical methods for implementing the operating system structure? Describe their characteristics.
2. What are the two methods for implementing the LRU Page Replacement algorithm? Describe their characteristics.

**Counter implementation:**

Every page table entry has a counter; every time page is referenced through this entry, copy the clock into the counter

When a page needs to be replaced, look at the counters to find smallest value

Search through table needed

**Stack implementation:**

Keep a stack of page numbers in a double link form:

**Page referenced:**

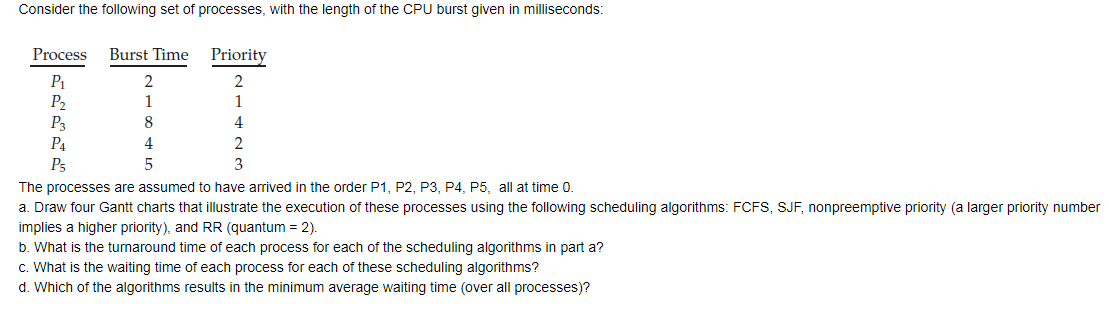
move it to the top, the most recently used page is always at the top of the stack and the least recently used page is always at the bottom

requires 6 pointers to be changed at worse

But each update more expensive

No search for replacement

综合题提纲：

1. 

（复习PPT的P91）

1. In a page virtual storage system, supposes that a process is assigned with 4 physical frames. In the beginning the memory is empty. The sequence string for accessing the pages is 7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 7, 1, 4, 5, 4, 1, 2, 3, 1. Write down the number of page faults when using FIFO and LRU two page replacement algorithms, and draw the page trend graph accordingly. (复习PPT的P237)

3. (复习PPT的P155)

