西 安 电 子 科 技 大 学

考试时间 120 分钟

试　　题

题号	一	二	三	四	总分
分数					

1. 考试形式：闭卷；2。考试日期　　年　月　日 3. 本试卷共四大题，满分 100 分。

Part1: Select one answer (A through D) for each question (Total 10 points, each 1)

1. For four conditions listed below to have a good solution of race conditions, which one is not correct?

A. No two processes may be simultaneously (同时) outside their critical regions (临界区).

B. No assumptions (假设) may be made about speeds or the number of CPUs.

C. No process running outside its critical region may block other processes.

D. No process should have to wait forever to enter its critical region.

2. What is the difference between C library function “fread” and system call “read”?

A. fread could read file and read can't　B. fread has less number of parameters

C. read has less instructions than fread.　D. read has more instructions than fread.

3. For memory-mapping I/O, which statement is correct？.

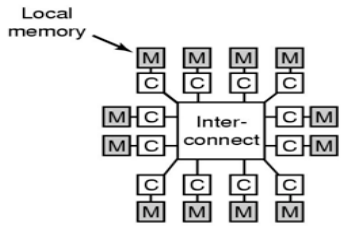
A. Memory-mapping I/O can't access I/O ports

B. Memory-mapping I/O can't access files in file system.

C. Memory-mapping I/O can access I/O ports

D. Memory-mapping I/O can only access executable files in file system

4. The Multiprocessor Systems diagram listed below is a _____



A. shared memory model **B. message passing multi-computer**
C. wide area distributed system **D. None of the above**

5. If the banker algorithm is used in a system, which of the following statement is correct?_____

A. When the system is in an unsafe state, it is a deadlock state.
B. When the system is in an unsafe state, deadlock may occur.
C. When the system is in a safe state, it is a deadlock state
D. When the system is in a safe state, it will definitely reach a deadlock state.

6. When a process is in the running state, ____.

A. it's waiting for the CPU
B. it is in the ready queue
C. it's in the waiting queue
D. part of its code is in the CPU.

7. For the variable partition (划分) allocation scheme, after a job is completed, the system will reclaim (回收) its main memory space and merge it with the adjacent (相邻) free area. In the following situation, which one will decrease (减少) the number of free area by 1 after memory reclamation? ____

A. no upper free area and no lower free area
B. one upper free area and no lower free area
C. no upper free area and one lower free area
D. one upper free area and one lower free area

8. In the system of segment storage management, if the address is represented by 24 bits, and 8 bits of them are segment numbers, the maximum length of each segment is

allowed to be _____.

A.2^{24} B. 2^{16} C.2^{8} D.2^{32}

9. The output speed of CPU is much faster than that of printer. To solve this problem, we can use _____

A. Parallel technology

B. Buffer technology

C. Virtual memory technology

D. Coverage technology

10. From the user's point of view, the operating system is _____.

A. Extended machine, i.e. providing an abstract interface between user and computer hardware.

B. Software for limiting the speed of processes.

C. Software for organizing computer workflow reasonably

D. A set of resources

Part2: Fill Blanks (Total 10 points, each 1)

1. Security Goals includes: _____ , _____ , _____.

2. Shared library is also called _____.

3. TLB means _____.

4. Process is an abstraction (抽象) of a _____ program.

5. A lock that uses _____ is called a spin (自旋) lock.

6. Because threads have some of the properties of processes, they are sometimes called _____ processes.

7. Process States includes: new, running, _____, _____ and terminated.

Part3: Essay Questions (Total 30 points, each 6)

1. What's the difference between semaphore and busy-waiting lock?

2. For the program listed below, how many "hello" will be printed? Please explain your answer.

```
int main()
{
    fork();
    printf("hello\n");
    execl("/bin/ls","ls","-l",0);
    printf("hello\n");
}
```

3. For the code of busy waiting lock listed below, (1) please fill the blanks & write down the initial (初始) value of variable LOCK. (2) And then explain the semantics of the statement:"XCHG REGISTER,LOCK". (3) Further, please explain why this statements can't be replaced by the code: "MOVE REGISTER2 REGISTER; MOVE REGISTER LOCK; MOVE LOCK REGISTER2".

```
enter region:
        MOVE REGISTER,  __(1)__
        XCHG REGISTER,LOCK
        CMP REGISTER,#0
        JNE enter region
```

```
leave_region:
    MOVE LOCK,  _(2)___
    RET
```

4. Please show an example that when circular wait condition occurs, the deadlock doesn't occur.

5. For the quota table, why does it limit the number of files?

Part4: Integrate Questions (Total 50 points, each 10 points).

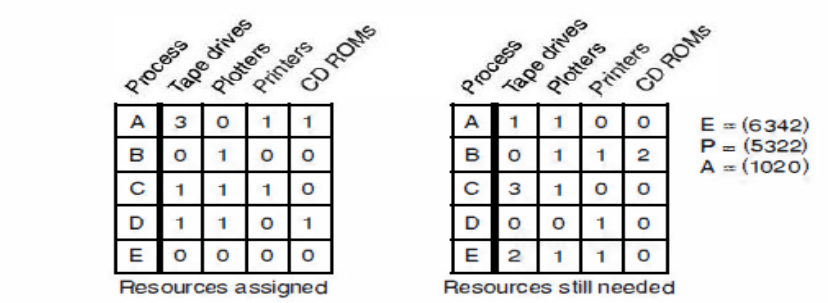
1. In a request paging system, assume that the number of physical pages allocated by the system to a process is 3, and the memory is empty at the beginning. Perform the following access page number sequence:

1，2，3，4，2，5，3，2，1，4，1，5

How many times are missing pages when using OPT(最优) and LRU for page replacement?

2.Please explain the difference of bitmap memory allocation and list memory allocation?

3. For the current resource allocation and requirement matrix and resource vectors（Vector E represents total resources, vector P represents allocated resource at present time, vector A represents currently available resources）listed below, is the current state a safe state? And explain why.



Process	Tape drives	Plotters	Printers	CD ROMs
A	3	0	1	1
B	0	1	0	0
C	1	1	1	0
D	1	1	0	1
E	0	0	0	0

Resources assigned

Process	Tape drives	Plotters	Printers	CD ROMs
A	1	1	0	0
B	0	1	1	2
C	3	1	0	0
D	0	0	1	0
E	2	1	1	0

Resources still needed

E = (6342)
P = (5322)
A = (1020)

4. What's the difference of Programmed I/O, Interrupt-Driven I/O, and I/O using DMA?

5. For the solution of Dining Philosophers Problem, please write the code of function "test ".

```
#define N               5           /* number of philosophers */
#define LEFT            (i+N−1)%N   /* number of i's left neighbor */
#define RIGHT           (i+1)%N     /* number of i's right neighbor */
#define THINKING        0           /* philosopher is thinking */
#define HUNGRY          1           /* philosopher is trying to get forks */
#define EATING          2           /* philosopher is eating */
typedef int semaphore;              /* semaphores are a special kind of int */
int state[N];                       /* array to keep track of everyone's state */
semaphore mutex = 1;                /* mutual exclusion for critical regions */
semaphore s[N];                     /* one semaphore per philosopher */

void philosopher(int i)             /* i: philosopher number, from 0 to N−1 */
{
    while (TRUE) {                  /* repeat forever */
        think( );                   /* philosopher is thinking */
        take_forks(i);              /* acquire two forks or block */
        eat( );                     /* yum-yum, spaghetti */
        put_forks(i);               /* put both forks back on table */
    }
}

void take_forks(int i)              /* i: philosopher number, from 0 to N−1 */
{
    down(&mutex);                   /* enter critical region */
    state[i] = HUNGRY;              /* record fact that philosopher i is hungry */
    test(i);                        /* try to acquire 2 forks */
    up(&mutex);                     /* exit critical region */
    down(&s[i]);                    /* block if forks were not acquired */
}

void put_forks(i)                   /* i: philosopher number, from 0 to N−1 */
{
    down(&mutex);                   /* enter critical region */
    state[i] = THINKING;            /* philosopher has finished eating */
    test(LEFT);                     /* see if left neighbor can now eat */
    test(RIGHT);                    /* see if right neighbor can now eat */
    up(&mutex);                     /* exit critical region */
}
```