1 INTRODUCTION

1.6 SYSTEM CALL

1. To a programmer, a system call looks like any other call to a library procedure. Is it important that a programmer know which library procedures result in system calls? Under what circumstances and why?

2 PROCESS AND THREAD

2.1 PROCESS

1. Please use program to explain the difference between fork and exec
2. How many processes will be created after the execution of the program below?

```
int main()
{
   pid_t pid;
   int i;
     for(i=0;i<2;i++)
     {
            pid=fork();
   }
}
```

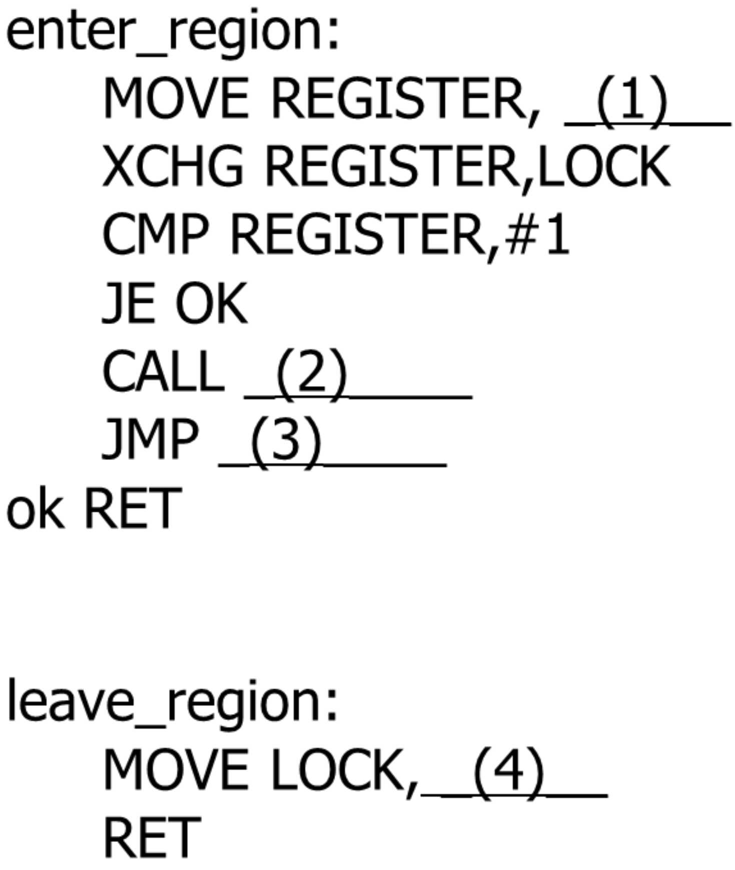
3. When a shell script is executed, will OS create a process for it? If so, according to the text book, which event cause the process to be created?
4.Plese use shell commands to demonstrate the process states and hierarchies.
5. What is PCB? What is "working directory" in PCB？Please use program to show the existence of "working directory". What is the difference between process and program?

2.2 THREADS

1. Please write a program that creates a thread, then answer the following questions:
i．explain why thread is proposed when the concept “PROCESS” has been existed?
ii．What is the pros&cons of user level thread? What about kernel level thread?
iii . When you running the program in our Kylin experimental platform, for the thread you create in your program, is it a user level thread or a kernel level thread?

2.3 IPC

1. For the mutex implementation program below, please fill the blanks and then specify the initial value of LOCK, finally explain why the initial value of LOCK should be the one you give.



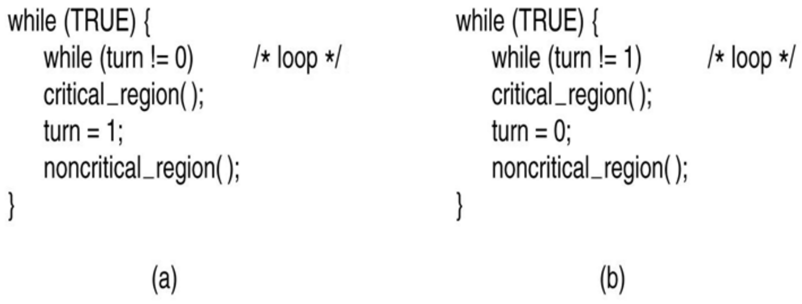
```
enter_region:
     MOVE REGISTER, _(1)__
     XCHG REGISTER,LOCK
     CMP REGISTER,#1
     JE OK
     CALL _(2)_____
     JMP _(3)_____
ok RET


leave_region:
     MOVE LOCK,_(4)__
     RET
```

2.What is race condition? What is mutual exclusion? What is critical region? Please use a program to demonstrate race condition, then use mutual exclusion mechanism to protect the critical region, avoiding the race condition.

3. What's the difference between mutex and semaphore?
4. What are four conditions to hold to have a good solution for race condition/mutual

exclusion? For the condition, "No process running outside its critical region may block other processes.", please explain why this condition is correct.

5.For the strict alternation given below, if there are some errors of this program, please correct it. If there is no error, please explain why the strict alternation can be done.



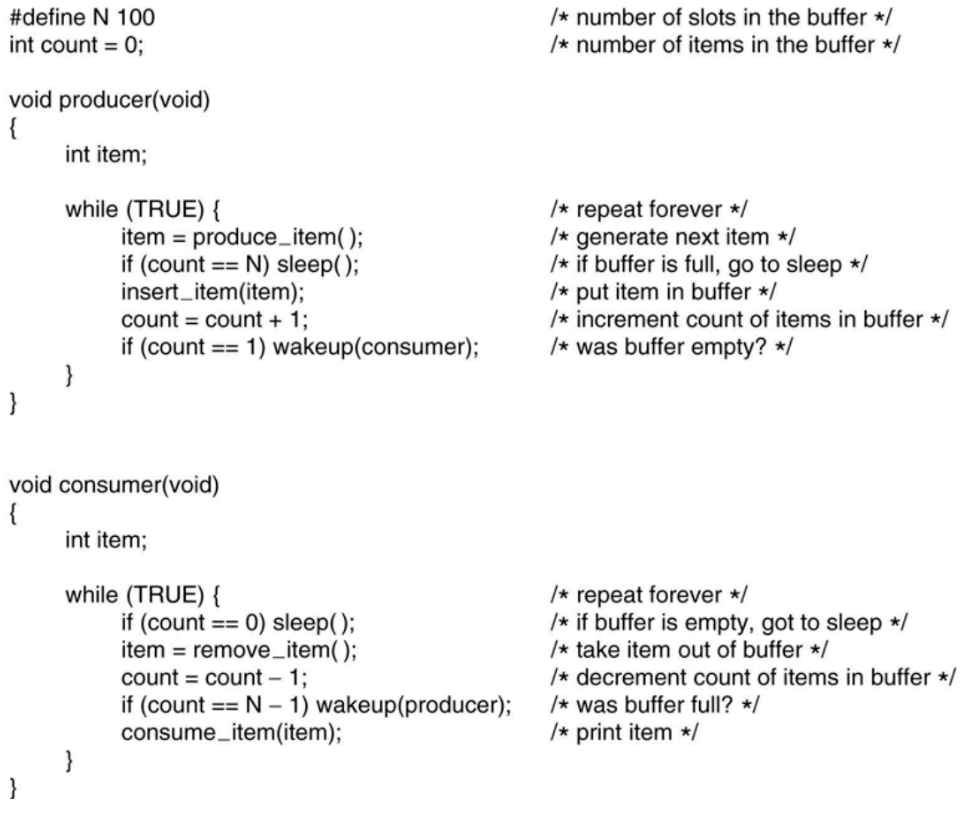
```
while (TRUE) {
    while (turn != 0)        /* loop */
    critical_region( );
    turn = 1;
    noncritical_region( );
}
```

(a)

```
while (TRUE) {
    while (turn != 1)        /* loop */
    critical_region( );
    turn = 0;
    noncritical_region( );
}
```

(b)

6. **What are errors of the code listed below?**



```
#define N 100                                   /* number of slots in the buffer */
int count = 0;                                  /* number of items in the buffer */

void producer(void)
{
     int item;

     while (TRUE) {                             /* repeat forever */
          item = produce_item( );               /* generate next item */
          if (count == N) sleep( );             /* if buffer is full, go to sleep */
          insert_item(item);                    /* put item in buffer */
          count = count + 1;                    /* increment count of items in buffer */
          if (count == 1) wakeup(consumer);     /* was buffer empty? */
     }
}


void consumer(void)
{
     int item;

     while (TRUE) {                             /* repeat forever */
          if (count == 0) sleep( );             /* if buffer is empty, got to sleep */
          item = remove_item( );                /* take item out of buffer */
          count = count − 1;                    /* decrement count of items in buffer */
          if (count == N − 1) wakeup(producer); /* was buffer full? */
          consume_item(item);                   /* print item */
     }
}
```