國立中央大學資訊工程學系博士班105學年度第一學期資格考試題紙

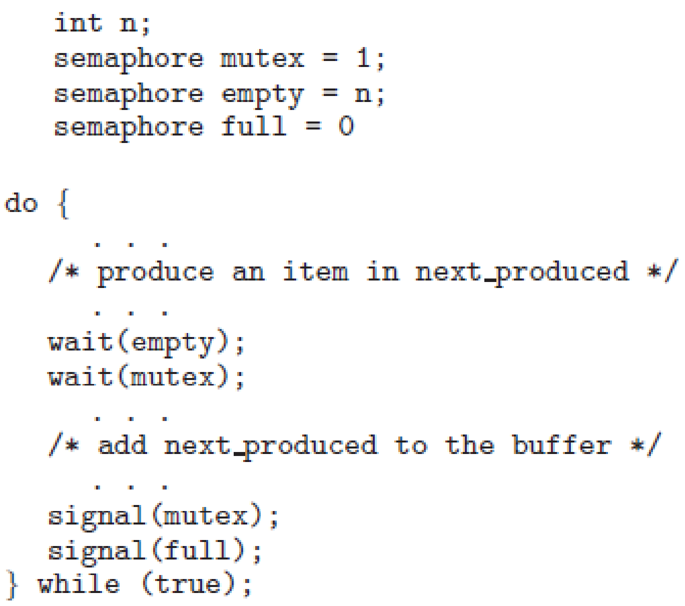
科目：作業系統 (Operating System)　　第一頁　共四頁（page 1 of 4）

1. (10%)How does DMA increase system concurrency? How does it complicate the hardware and system design?
2. (10%)What problems could occur if a system allowed a file system to be mounted simultaneously at more than one location?
3. In the consumer-producer example program, a ring buffer queue is used to store the produced item that will be take off by the consumer later.

   There are two example program, the following one (figure 5.9, 5.10) uses semaphore to implement it. (p240, Fig 6.10/6.11　8th edition)

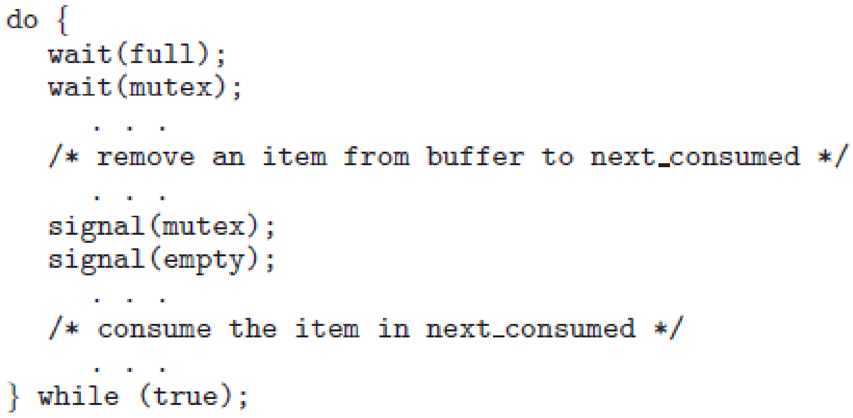
   3-a.(5%)What are the shared variables in figure 5.9, 5.10?

   3-b.(5%)Why this program need to use semaphore primitive ?



```
    int n;
    semaphore mutex = 1;
    semaphore empty = n;
    semaphore full = 0

do {
     . . .
   /* produce an item in next_produced */
     . . .
   wait(empty);
   wait(mutex);
     . . .
   /* add next_produced to the buffer */
     . . .
   signal(mutex);
   signal(full);
} while (true);
```

Figure 5.9　The structure of the producer process.

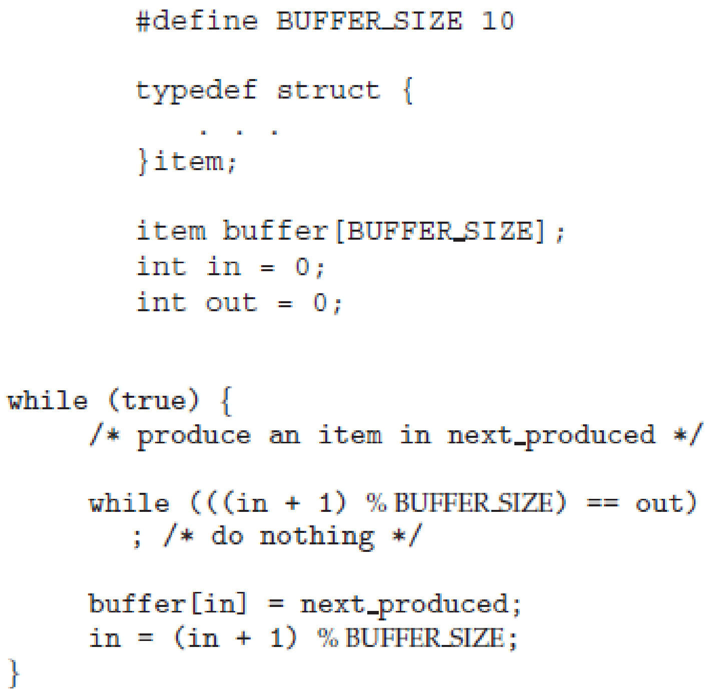


```
do {
   wait(full);
   wait(mutex);
      . . .
   /* remove an item from buffer to next_consumed */
      . . .
   signal(mutex);
   signal(empty);
      . . .
   /* consume the item in next_consumed */
      . . .
} while (true);
```

Figure 5.10 The structure of the consumer process.

3-c. (5%)The following figure 3.13 and 3.14 are another example program. What are the shared variables in these producer and consumer program?

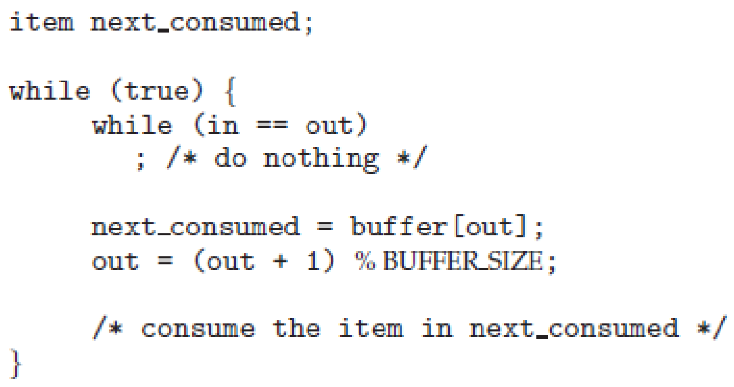
3-d. (5%) In what conditions the program need NOT to use the lock synchronization primitive to support the correct processing? (p118, fig 3.14/3.15　8th edition)



```
#define BUFFER_SIZE 10

typedef struct {
     . . .
}item;

item buffer[BUFFER_SIZE];
int in = 0;
int out = 0;
```

```
while (true) {
     /* produce an item in next_produced */

     while (((in + 1) % BUFFER_SIZE) == out)
        ; /* do nothing */

     buffer[in] = next_produced;
     in = (in + 1) % BUFFER_SIZE;
}
```

Figure 3.13 The producer process using shared memory.



```
item next_consumed;

while (true) {
     while (in == out)
        ; /* do nothing */

     next_consumed = buffer[out];
     out = (out + 1) % BUFFER_SIZE;

     /* consume the item in next_consumed */
}
```

Figure 3.14 The consumer process using shared memory.

3-e. (5%) If kernel monitor(R.C.A. Hoare Monitor) approach is adopted to support producer and consumer program. Any processes that can not call consumer or producer function in monitor simultaneously. That means the monitor restricts the parallel processing of these two functions. What is your suggestion to support multiprocessor parallel processing?

3-f. (5%). Prior to version 2.6, Linux was a nonpreemptive kernel. What is the approach used in Linux 2.6 ?

4. (30%) Assume you have a system with a static priority CPU scheduler. Assume the scheduler supports preemption. Describe what the program below prints in sequence, where the priorities are set such that T2 has a high priority, T1 has the middle priority, and T0 has the low priority. Assume the system starts with only T0 executing. Assume the semaphore mutex is initialized to 1.

```
void T0 ( )   {
   printf ( “T0-Start \n” );
   StartThread (T1);
   printf ( “T0-End \n” );
}
```

```
void T1 ( )   {
   printf ( “T1-Start \n” );
   P (mutex);
   printf ( “T1-A \n” );
   StartThread ( T2 );
   printf ( “T1-B \n” );
   V (mutex);
   printf ( “T1-End \n” );
}
```

```
void T2 ( )   {
   printf ( “T2-Start \n”);
   P (mutex);
   printf ( “T2-A \n” );
   V (mutex);
   printf ( “T2-End \n” );
}
```

5. (20%) Are the following statements about IP addresses true or false? Be sure to justify your answer for false statements. For each statement, you will get 4 points for correct answer, zero point for blank, or -2 point for incorrect answer.

    (a) Domain Name Service (DNS) can be used to acquire IP addresses.

    (b) Address Resolution Protocol (ARP) can be used to acquire IP addresses.

    (c) Network Address Translation (NAT) is used to map MAC addresses to IP addresses.

    (d) IP Multicasting is adopted in Dynamic Host Configuration Protocol (DHCP).

    (e) IPv6 addresses are 128 bits long.