# 國立中山大學資訊工程學系

## ９２學年度第１學期博士班資格考試　作業系統

1. (10 marks) Consider a system with four processes $P_1$ through $P_4$ and five allocatable resources $R_1$ through $R_5$. The current allocation and maximum needs are as follows:

| | $Allocated$ | $Maximum$ | $Available$ |
|---|---|---|---|
| $P_1$ | 1 0 2 1 1 | 1 1 2 1 3 | 0 0 $x$ 1 1 |
| $P_2$ | 2 0 1 1 0 | 2 2 2 1 0 | |
| $P_3$ | 1 1 0 1 0 | 2 1 3 1 0 | |
| $P_4$ | 1 1 1 1 0 | 1 1 2 2 1 | |

   What is the smallest value of $x$ for which this is a safe state? For full credit, justify your answer.

2. (16 marks total; 4 marks each subquestion) Consider the following set of processes, with the length of the CPU-burst time given in milliseconds:

| $Process$ | $BurstTime$ | $Priority$ |
|---|---|---|
| $P_1$ | 10 | 3 |
| $P_2$ | 1 | 1 |
| $P_3$ | 2 | 3 |
| $P_4$ | 1 | 4 |
| $P_5$ | 5 | 2 |

   The processes are assumed to have arrived in the order $P_1$, $P_2$, $P_3$, $P_4$, $P_5$, all at time 0.

   (a) Draw four Gantt charts illustrating the execution of these processes using FCFS, SJF, a nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum=1) scheduling.

   (b) What is the turnaround time of each process for each of the scheduling algorithms in part (a)?

   (c) What is the waiting time of each process for each of the scheduling algorithms in part (a)?

   (d) Which of the schedulers in part (a) results in the minimal average waiting time (over all processes)?

3. (12 marks total; 4 marks each subquestion) Consider the following page reference string:

   1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

   How many page faults would occur for the following replacement algorithms, assuming one, two, three, four, five, six, or seven frames? Remember that all frames are initially empty, so your first unique pages will all cost one fault each.

   (a) LRU replacement

   (b) FIFO replacement

   (c) Optimal replacement

4. (12 marks total; 2 marks each subquestion) Suppose that a disk drive has 5000 cylinders, numbered from 0 to 4999. The drive is currently serving a request at cylinder 143, and the previous request was at cylinder 125. The queue of pending requests, in FIFO order, is

   86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130.

Starting from the current head position, what is the schedule that the disk arm moves to satisfy all the pending requests for each of the following disk scheduling algorithms?

(a) FCFS
(b) SSTF
(c) SCAN
(d) LOOK
(e) C-SCAN
(f) C-LOOK

5. (5 marks) In a ring-protection system, level 0 has the greatest access to objects, and level $n$ (greater than zero) has fewer access rights. The access rights of a program at a particular level in the ring structure are considered as a set of capabilities. What is the relationship between the capabilities of a domain at level $j$ and a domain at level $i$ to an object (for $j > i$)?

6. (10 marks total; 2 marks each subquestion) Measurements of a certain system have shown that the average process runs for a time $T$ before blocking on I/O. A process switch requires a time $S$, which is effectively wasted (overhead). For round-robin scheduling with quantum $Q$, give a formula for the CPU efficiency for each of the following:

   (a) $Q = \infty$
   (b) $Q > T$
   (c) $S < Q < T$
   (d) $Q = S$
   (e) $Q$ nearly 0

7. (10 marks total; 5 marks each subquestion) Suppose that two processes, $P_1$ and $P_2$, are running in a uniprocessor system. $P_1$ has two threads. $P_2$ has three threads. All threads in both processes are CPU-intensive; that is, they never block for I/O. The operating system uses simple round-robin scheduling.

   (a) Suppose that all of the threads are user-level threads, and that user-level threads are implemented using a single kernel thread per process. What percentage of the processor's time will be spent running $P_1$'s threads? For full credit, justify your answer.
   (b) Suppose instead that all of the threads are kernel threads. What percentage of the processor's time will be spent running $P_1$'s threads? For full credit, justify your answer.

8. (9 marks total; 3 marks each subquestion) What is the cause of thrashing? How does the system detect thrashing? Once it detects thrashing, what can the system do to eliminate this problem?

9. (16 marks total; 4 marks each subquestion) Consider a variation of the bounded buffer producer/consumer problem in which there are many concurrent producers and the items produced and consumed vary in size. The total capacity of the buffer is $N$. These processes use two semaphores to synchronize access to the buffer: `full` is a counting semaphore with initial value 0 and `empty` is a counting semaphore with initial value $N$. Each producer executes the following code when it wishes to insert an item of size $k$ ($1 \leq k \leq N$) into the buffer:

```
for i from 1 to k do {
  P(empty)
}
insert item of size k into buffer
V(full)
```

The consumer executes the following code when it wishes to remove an item from the buffer:

```
P(full)
remove item of size k from the buffer
for i from 1 to k do {
  V(empty)
}
```

In each of the code fragments shown above, $i$ is an integer variable that is local to each process. Answer each of the following questions. In each case, you must justify your answer to receive full credit.

(a) Does the code above ensure that producers do not insert items into the buffer unless the buffer has room to hold them?

(b) Does the code above ensure that the consumer does not remove items from the buffer unless the buffer contains items to remove.

(c) Does the code above ensure that only one process (producer or consumer) will use (insert or remove items from) the buffer at a time?

(d) Does the code above ensure that processes using the buffer will not deadlock?