國立中山大學資訊工程學系

105 學年度第 1 學期博士班資格考試

科目：作業系統

INSTRUCTIONS: *If any question is unclear or you believe some assumptions need to be made, state your assumptions clearly at the beginning of your answer.*

---

1. (10%) What would be the output of the following C program? (*Note that the line numbers are for reference only.*)

```
 1 #include <stdio.h>
 2 #include <stdlib.h>
 3 #include <unistd.h>
 4 #include <pthread.h>
 5 #include <sys/types.h>
 6 #include <sys/wait.h>
 7 static void *runner(void *param)
 8 {
 9     (* (int*) param)--;
10     pthread_exit(0);
11 }
12 int main(int argc, char **argv)
13 {
14     int value = 11;
15     pid_t pid = fork();
16     if (pid > 0) {
17         int status;
18         waitpid(-1, &status, 0);
19         printf("A = %d\n", ++value);
20     }
21     else if (pid == 0) {
22         pid_t pid = fork();
23         if (pid > 0) {
24             int status;
25             waitpid(-1, &status, 0);
26             printf("B = %d\n", value++);
27         }
28         else if (pid == 0) {
29             pid_t pid = fork();
30             pthread_t tid;
31             pthread_create(&tid, NULL, runner, &value);
32             pthread_join(tid, NULL);
33             if (pid > 0) {
34                 int status;
35                 waitpid(-1, &status, 0);
36                 printf("C = %d\n", ++value);
37             }
38             else
39                 printf("D = %d\n", value++);
40         }
41         else
42             return 1;
43     }
44     else
45         return 1;
46     return 0;
47 }
```

2. (10%; 5% each) Assume a page reference string for a process with m frames (initially all empty). The page reference string has length p with q distinct page numbers occurring in it. For any page-replacement algorithms,

   (a) What is an upper bound on the number of page faults? **Justify your answer for credit.**

   (b) What is a lower bound on the number of page faults? **Justify your answer for credit.**

3. (10%; 5% each) Given an i-node with sixteen direct blocks and four levels of indirect blocks and assuming that the sizes of a pointer and a block are, respectively, 8 bytes and 8 Kbytes,

   (a) What would be the size of the smallest file allowed in blocks?

   (b) What would be the size of the largest file allowed in blocks?

4. (10%; 5% each) A disk has 12000 cylinders, each with 16 tracks of 512 blocks. A seek takes 1 ms per cylinder moved. If no attempt is made to put the blocks of a file close to each other, two blocks that are logically consecutive (i.e., follow one another in the file) will require an average seek, which takes 6 ms. If, however, the operating system makes an attempt to cluster related blocks, the mean interblock distance can be reduced to 2 cylinders and the seek time reduced to 200 μs. How long does it take to read a 100 block file in both cases, if the rotational latency is 2 ms and the transfer time is 20 μs per block?

5. (10%) A computer whose processes have 1024 pages in their address spaces keeps its page tables in memory. The overhead required for reading a word from the page table is 1100 nsec. To reduce this overhead, the computer has a TLB, which holds 32 (page, frame) pairs and can do a lookup in 100 nsec. What hit rate is needed to reduce the mean overhead to 200 nsec?

6. (10%) A machine has 48-bit virtual addresses and 32-bit physical addresses. Pages are 4 KB. How many entries are needed for the page table?

7. (10%; 5% each) Consider the interprocess-communication scheme where mailboxes are used. Suppose a process P wants to wait for two messages, one from mailbox A and one from mailbox B. What sequence of `send` and `receive` should it execute so that the messages can be received in any order?

8. (10%; 5% each) Consider the two-dimensional array `a`:

```
double a[][] = new double[200][200];
```

   where each double occupies 8 bytes and `a[0][0]` is at location 200, in a paged system with pages of size 200 bytes. A small process is in page 0 (locations 0 to 199) for manipulating the matrix; thus, every instruction fetch will be from page 0. For three page frames, how many page faults are generated by the following array initialization loops, using LRU replacement and assuming (1) page frame 0 has the process in it, (2) the other two are initially empty, and (3) the array is stored in memory row-major.

   (a)
```
for (int i = 0; i < 200; i++)
    for (int j = 0; j < 200; j++)
        A[i][j] = 0;
```

   (b)
```
for (int j = 0; j < 200; j++)
    for (int i = 0; i < 200; i++)
        A[i][j] = 0;
```

9. (10%; 5% each) Suppose that a scheduler has k ready processes at time 0, and that no new processes are created after time 0. Process i ($0 < i \leq k$) requires i units of computing time. Answer each of the following questions.

   (a) For a preemptive, round-robin scheduler with a scheduling quantum of one time unit, what is the mean turnaround time for these processes, assuming that process k is at the front of the ready queue and that other processes appear in decreasing order of required computing time?

   (b) For a non-preemptive, shortest-job-first scheduler, what is the mean turnaround time for these processes?

10. (10%; 5% each) Suppose that a disk drive has 1000 cylinders, numbered from 0 to 999. The drive is currently serving a request at cylinder 200, and the previous request was at cylinder 125. The queue of pending requests, in FIFO order, is

   50, 500, 250, 800, 350, 550, 400, 600, 100.

   Starting from the current head position, what is the *total distance* (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk scheduling algorithms?

   (a) C-SCAN

   (b) C-LOOK