Operating Systems, Spring 2019

Midterm

2:10pm ~ 3:50pm, Friday, April 19, 2019

INSTRUCTIONS:

1. This is a *closed-book* exam.
2. Try to solve all of the problems.
3. Try to give short answers. (Hint: An answer need not always be longer than the question.)
4. No cheating.
5. Please hand in both the exam sheet and the answer sheet.
6. Please note that unless otherwise stated, all the line numbers for the program listings are for reference only.
7. 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. (20%) What would be the output of the following C program that uses the Pthreads API?

```
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
8  static void *runner(void *p)
9  {
10     --(*(int*)p);
11     pthread_exit(0);
12 }
13
14 int main(int argc, char **argv)
15 {
16     int value = 99;
17     pid_t pid = fork();
18     if (pid > 0) {
19         waitpid(-1, NULL, 0);
20         printf("A = %d\n", --value);
21     }
22     else if (pid == 0) {
23         pid_t pid = fork();
24         if (pid > 0) {
25             waitpid(-1, NULL, 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                 waitpid(-1, NULL, 0);
35                 printf("C = %d\n", ++value);
36             }
37             else {
38                 printf("D = %d\n", value++);
39             }
40         }
41         else {
42             return 1;
43         }
44     }
45     else {
46         return 1;
47     }
48     return 0;
49 }
```

2. (20%) 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 (i.e., the useful CPU time divided by the total CPU time) for each of the following:

   (a) $Q = \infty$

   (b) $Q > T$

   (c) $S < Q < T$

   (d) $Q = S$

   To simplify the answers, you may assume Q divides T evenly.

3. (20%) 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?

4. (10%) Use `typedef` in C to declare 'CMP' as a pointer to function taking as input two generic pointers pointing to something that cannot be modified and returning as output an integer.

5. (10%) What would be the output of the following C program, assuming `int` occupies 4 bytes?

```
1 #include <stdio.h>
2
3 int a[10][30][60];
4
5 int main()
6 {
7     printf("%d\n", (int) (a[5]-a[15]));
8     return 0;
9 }
```

6. (10%) What would be the output of the following C program, assuming `int` occupies 4 bytes?

```
1 #include <stdio.h>
2
3 int a[10][30][60];
4
5 int main()
6 {
7     printf("%d\n", (int) sizeof(a[5]));
8     return 0;
9 }
```

7. (10%) What would be the output of the following C program? Why?

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int a = 10;
6     printf("a=%d\n", ++a--);
7     return 0;
8 }
```