

Full Name:.....

Roll Number:.....

IT215: Systems Software, Winter 2016-17
First In-Sem Exam (1 hour)

Feb 7, 2017

Instructions:

- Make sure your exam is not missing any sheets, then write your name and roll number on the top of this page.
- Clearly write your answer in the space indicated. None of the questions need long answers.
- For rough work, do not use any additional sheets. Rough work will not be graded.
- Assume IA32 machine running Linux.
- The exam has a maximum score of 20 points. It is CLOSED BOOK. Notes are NOT allowed.
- Anyone who copies or allows someone to copy will receive F grade.

Problem 1 (8):
Problem 2 (/2):
Problem 3 (/2):
Problem 4 (/4):
Problem 5 (/4):
TOTAL (/20):

Problem 1. (8, 2 each points):

Circle the *single best* answer to each of the following questions. You will get -1 points for a wrong answer. If you circle more than one answer, you will lose the mark for the corresponding question. Assume that all system calls execute without error. Additionally, assume that each call to `printf` flushes its output just before it returns.

1. How many times will the following program print "Hello"?

```
int main() {
    int i;
    for (i = 0; i < 2; i++) {
        fork();
        printf("Hello\n");
    }
    return 0;
}
```

- (a) 2 (b) 4 (c) 6 (d) 8

2. Which ONE of the choices below is a possible output of the following program?

```
pid_t pid;
int counter = 2;

void handler(int sig) {
    counter = counter - 1;
    printf("%d", counter); fflush(stdout);
    exit(0);
}

int main()
{
    signal(SIGUSR1, handler);
    printf("%d", counter); fflush(stdout);
    if ((pid = fork()) == 0)
        while (1) ;
    kill(pid, SIGUSR1);
    waitpid(-1, NULL, 0);
    counter = counter + 1;
    printf("%d", counter);
    exit(0);
}
```

- (a) 231
(b) 211
(c) 233
(d) 213

3. The file descriptor table

- (a) is identical in the parent and the child immediately after a fork
(b) is unchanged after a call to one of the exec functions
(c) Both (a) and (b) are correct
(d) Neither (a) nor (b) is correct

4. Given the following program:

```
int main() {
    int n = 1;
    if (fork() == 0) {
        n = n * 2;
        printf("%d, ", n);
        n = n * 2;
    }
    if (fork() == 0)
        n = n + 700;
    printf("%d, ", n);
    return 0;
}
```

Which of the following is *not* a possible output of the program?

- (a) 1, 2, 4, 704, 701,
- (b) 1, 704, 2, 4, 701,
- (c) 2, 4, 1, 701, 704,
- (d) 2, 704, 4, 701, 1,

Problem 2. (2 points):

Fill in the if-condition with the simplest correct boolean expression so that the statement printed by the program fragment is true.

```
if (          ) printf("I am not an orphan process\n");
```

Problem 3. (2, 1 each points):

Consider the following program.

```
int main() {
    int fd[2];
    pid_t pid;
    char c;

    pipe(fd);
    if ((pid = fork()) == 0) {
        printf("Child\n");
        write(fd[1], "a", 1);
        exit(0);
    }
    read(fd[0], &c, 1);
    printf("Parent\n");
}
```

Assuming no errors occur, circle the correct answer. Correct answers receive +1 mark, incorrect answers -0.5 mark, unanswered parts receive 0.

- | | | |
|------|-------|--|
| TRUE | FALSE | It is possible for "Child" to be printed before "Parent" |
| TRUE | FALSE | It is possible for "Parent" to be printed before "Child" |

Problem 4. (4, 2 each points):

Consider the following code, which has been written with the assumption that an unpredictable number of SIGINT interrupts can arrive asynchronously.

```
1  int i = 1;
2
3  void handler(int sig) {
4      i = 0;
5  }
6
7  int main() {
8      int j;
9      sigset_t s;
10     sigemptyset(&s);
11     sigaddset(&s, SIGINT);
12     signal(SIGINT, handler);
13     for (j = 0; j < 100; j++) {
14         i++;
15     }
16     printf("i = %d\n", i);
17     exit(0);
18 }
```

Now consider the following values for `i` that may be printed by the `printf` command: 0, 1, 100, 101.

For each question, indicate if and where the given calls to `sigprocmask` need to be inserted in order to obtain precisely the indicated set of possible outputs among 0, 1, 100, 101. Note that any given run prints just one value, and that the program may also print other values, but we are only interested in 0, 1, 100, and 101.

1. Just 101, but not 0, 1, or 100.

- (a) Neither needs to be inserted (leave next two questions blank).
- (b) Insert `sigprocmask(SIG_BLOCK, &s, 0);` right after line ____.
- (c) Insert `sigprocmask(SIG_UNBLOCK, &s, 0);` right after line ____.

2. Just 100 and 101, but not 0, 1.

- (a) Neither needs to be inserted (leave next two questions blank).
- (b) Insert `sigprocmask(SIG_BLOCK, &s, 0);` right after line ____.
- (c) Insert `sigprocmask(SIG_UNBLOCK, &s, 0);` right after line ____.

Problem 5. (4 points):

You are given two text files `letter.txt` and `number.txt`. Suppose `letter.txt` contains `abcdefghijklmnopqrstuvwxyz` and `number.txt` contains `0123456789`. Consider the following code. (We don't check return values for space reasons. Assume all functions return normally.) Write down the values of `ch` in the comments (an example answer is given at line 13).

```
1  int main(int argc, char *argv[])
2  {
3      int fd1, fd2, fd3, fd4;
4      char ch;
5
6      fd1 = open("number.txt", O_RDONLY);
7      fd2 = open("number.txt", O_RDONLY);
8
9      fd3 = open("letter.txt", O_RDONLY);
10     fd4 = dup(fd3);
11
12     read(fd1, &ch, 1);
13     printf("%c", ch);           /* ch = '0' */
14
15     read(fd2, &ch, 1);
16     printf("%c", ch);           /* ch = ____ */
17
18     read(fd3, &ch, 1);
19     printf("%c", ch);           /* ch = ____ */
20
21     read(fd4, &ch, 1);
22     printf("%c", ch);           /* ch = ____ */
23     fflush(stdout);
24
25     if (fork() == 0) {
26         read(fd1, &ch, 1);
27         printf("%c", ch);       /* ch = ____ */
28
29         exit(0);
30     }
31     wait(NULL);
32
33     read(fd1, &ch, 1);
34     printf("%c", ch);           /* ch = ____ */
35     fflush(stdout);
36
37     return 0;
38 }
```

(Blank page for rough work.)