2007-2008 FALL - SYSTEMS PROGRAMMING

FINAL EXAM **Jan. 10, 2008**

1. *(25 pts)* Part of a C language project can be implemented in Assembly either by calling Assembly functions from C or by using inline Assembly. Answer the following questions and explain your answers.
   (a) Which approach is expected to execute faster?
   (b) Considering the size of the Assembly portion, which approach should be preferred in which case?
   (c) Which approach is easier to compile and link?
   (d) Which approach is easier to maintain and modify?

2. *(25 pts)* Consider the piece of code given below. Assume all required definitions and declarations have been done. (Note: For simplicity, no error controls have been included.)
   (a) Explain how the program works considering all possible scenarios. (Use the line numbers in your explanation.)
   (b) Explain the system calls used in the program, including the purpose and value assignment of their input and output parameters.

```
1:  while (1) {
2:    FD_ZERO(&readfds);
3:    FD_SET(STDIN_FILENO, &readfds);
4:    FD_SET(fd1, &readfds);
5:    FD_SET(fd2, &readfds);
6:    tv.tv_sec = 5;
7:    tv.tv_usec = 0;
8:    select(f, &readfds, NULL, NULL, &tv);
9:    flag = 0;
10:   if (FD_ISSET(STDIN_FILENO, &readfds)) {
11:     i = read(STDIN_FILENO, inbuf, 1024);
12:     write(STDOUT_FILENO, inbuf, i);
13:     flag = 1;
14:   }
15:   if (FD_ISSET(fd1, &readfds)) {
16:     i = read(fd1, inbuf, 1024);
17:     write(STDOUT_FILENO, inbuf, i);
18:     flag = 1;
19:   }
20:   if (FD_ISSET(fd2, &readfds)) {
21:     i = read(fd2, inbuf, 1024);
22:     write(STDOUT_FILENO, inbuf, i);
23:     flag = 1;
24:   }
25:   if (!flag)
26:     break;
27: }
```

3. *(30 pts)* We want to write a character device driver that will act as an endless stream where a single character is continuously repeated. For example, if the character is 's', an n-byte read will produce an n-byte long stream of 's' characters. The character to repeat will be a module parameter with the default value 'a' and it will be modifiable through an ioctl command. Assume the major and minor numbers for the driver to be 250 and 1, respectively.
   (a) Explain the global variables which would be needed.
   (b) Write the module initialization function.
   (c) Write the read function.
   (d) Write the ioctl function that will change the character to repeat.
   (e) Give an example of a user-space program which demonstrates the use of these system calls.

4. *(20 pts)* Explain the role and purpose of the *request queue* for a block device driver.