Systems Programming 1 – 300167

Tutorial and Lab Practice Six (follows Lecture Six).
This work will not be marked. You should complete it within one week.

Tutorial

1. Read chapter five and six of the text book.

2. Review the terminology introduced and concepts taught in lecture 6.

3. What's the difference between a device file and a disk file? What's the difference between a connection to a device file and a connection to a disk file?

4. What is the difference between `ioctl()` and `fcntl()` system calls?

5. What is *canonical mode?* How does it work?

6. Name the four groups of *tty* driver settings, explain the purpose of each group, and name two bits in each group.

7. Test run program `rotate.c` (available in vUWS). Try using noncanonical mode by `stty -icanon; ./rotate`, compare the output and explain the logic.

Lab Practice

1. The textbook includes the following code to turn off disk buffering for a file descriptor. Write a function that turns buffering back on.

   ```
   #include <fcntl.h>
   int s;
   s = fcntl(fd,F_GETFL);
   s |=O_SYNC;
   result = fcntl(fd, F_SETFL, s);
   if (result==-1)
     perror("setting SYNC");
   ```

2. Modify the *rotate.c* program so it changes the tty mode itself. The revised program should turn off canonical mode and turn off echo. It should then read characters and print the next letter in the alphabet. When the user presses the letter "Q", the program should restore the tty setting and exit.

3. Modify the *sigdemo1.c* (available in vUWS) so it counts the number of time the user presses Ctrl-C. The revised program will print the message OUCH!, OUCH!!, OUCH!!!...where the number of the exclamation points equals the number of times the handler has been called. Also the program should accept an integer as command line argument. After the user presses Ctrl-C that many times, the program should exit.

Optional Work (For those students who wish to practise more.)

1. What is the effect on performance if you turn off buffering? Write a program that writes a large disk file in small piece, say a 1-megabyte file in 8-bytes chunks. Try this with O_SYNC on and with O_SYNC off. Adjust the file size and the chunk size to see how they affect the results.