Brian Duenas

CSE 460

Lab 3

20 points Total

1. **Replacing a Process Image**

**Q:** Modify test\_exec so that the function execl is used instead of using execlp.

**A:**



2. **Duplicating a Process Image**

**Q:** Try the "test\_fork.cpp" program and explain what you see on the screen.

**Output:**

**A:** The first line is the *cout* before the fork command. When the *fork* command is executed it creates a parent and child thread. Next the code run the *switch* statement, the child is told to print “This is the child” five times and the parent is told to print “This is the parent” three times. The print occur to whichever process is finished first.

3. **Waiting for a Process**

**Q:** Run the program and explain what you have seen on the screen.

**Output:**

**A:** Same output as before now with lines ten and eleven. At line ten the parent process is waiting for the child process to finish then prints line ten, and then prints exit code on line eleven of the child.

**Q:** Modify the program so that the child process creates another child and wait for it. The grand child prints out the id's of itself, its parent and grandparent.

**A:**

****

**Output:**

4. **Signals**

**Q:** Run the program and hit ^C for a few times. What do you see? Why?

**Output:**

**A:** The program print the results and when *CTRL+C* is pressed it prints the message in the function. *CTRL+*C is an interrupt so every time it happens it runs the function *func* instead of shutting the program down.

**Q:** Run "test\_alarm.cpp". What do you see? Why?

**Output:**

**A:** Alarm will wait five seconds before being set off by a signal. The signal is the child being killed.

**Q:** Modify your test\_signal.cpp program above by using sigaction() to intercept SIGINT; replace the "for" loop with "while ( 1 ); you should be able to quit the program by entering "^\". (Need to intercept SIGQUIT.)

**A:**



**Output:**



5. **Study of XV6**

**Q:** Compile and run xv6. Also run it in the debugger mode, disassemble the kernel in i386, and examine its code. Copy-and-paste some sample code in your report.

**A:**

Debugger Mode:

 GDB:



Broken *ls -l* command:

