CS590 Homework #2
Due: Thursday, June 24, 2010

1. Develop a program to provide the basic functionality of a UNIX shell

Your program will enter a command processing loop that does the following:

a. Display the prompt “shell> ”
b. Wait for the user to input a command
c. Parse the command
d. Check to see if the command is a builtin and handle it appropriately
e. If not a builtin, then execute the command using the same fork/exec method used by the shell.

Your program must support the following I/O redirection symbols for stdin (<), stdout (>), stdout with append (>>), and pipes (|).

Your program must support the background symbol (&) at the end of a command, and provide a signal handler for SIGCHLD.

Your shell must support the following builtin commands:

“showstatus”	to print the exit status of the last command.
“sleep <num>”	to pause for <num> seconds
“exit”	to terminate your shell

Note: The “sleep” builtin needs to make use of a SIGALRM handler

For example, a run of this program might be:

```
$  ./myshell
shell>  who > who.out
shell>  showstatus
shell>  sleep 10
shell>  ls –lsrt  | grep file3 >> ls.out
shell>  cat < file1 | wc-l
shell>  myprog arg1 > myprog.txt &
shell>  exit
```

To simplify the parsing of commands, you may assume that there will always be a space before and after an I/O redirection symbol.

For the purposes of this assignment, you may assume there will be no more than one pipe in a command, but your design should be flexible enough to provide for multiple pipes with some minor modifications.