Luke Jaffe
Operating Systems
Due 02/03/16

Homework #2

Question 2.

a)

Option 1 Method 1

Parent	**Child**	**Grandchild**
begin		
fork()	begin	
wait() // print	pipe(fds)	
	fork()	begin
	close(fds[0])	close(fds[1])
	close(1)	close(0)
	dup(fds[1])	dup(fds[0])
	exec(“ls”, ...)	exec(“wc”, ...)
finish	finish	finish

* Prints only when the child finishes

Option 1 Method 2

Parent	**Child**	**Grandchild**
begin		
pipe(fds1)		
fork()	begin	
	pipe(fds2)	
	fork()	begin
close(fds1[1])	close(fds1[0])	
read(grandchild_pid)	write(grandchild_pid)	
wait() // for child, print	close(fds2[0])	close(fds2[1])
waitpid(grandchild_pid) // print	close(1)	close(0)
	dup(fds2[1])	dup(fds2[0])
	exec(“ls”, ...)	exec(“wc”, ...)
finish	finish	finish

* Prints when child and grandchild finish

Option 2

Parent	**Child1**	**Child2**
begin		
pipe(fds)		
fork() x 2	begin	begin
close(fds[0])	close(fds[0])	close(fds[1])
close(fds[1])	close(1)	close(0)
wait() x 2	dup(fds[1])	dup(fds[0])
	exec(“ls”, ...)	exec(“wc”, ...)
finish	finish	finish

b)

Either option will work fine for piping the output of one command into the input of another; however, option 1 makes it harder to display a message after the grandchild process is finished executing. If the child process calls wait() before executing “ls -l”, then the program will hang, since the grandchild process executing “wc” waits for input before exiting. On the other hand, the child process can't execute any code after calling execvp(), since the process image is replaced by the “ls” program. Option 1 method 1 will pipe correctly, and print when the first child process finishes. Option 1 method 2 solves the problem of the child not being able to wait by passing the pid of the grandchild to the parent after forking in the child. Then the parent calls wait() once for the child, and waitpid() for the grandchild. This works, but is an inefficient and badly abstracted way of solving the problem.

For option 2, the main process can simply fork twice, and wait for both processes to finish. This way, no hackish pid passing is required, and no therefore no extra pipes are needed. The main problem with option 1 is that it doesn't solve the issue of a parent process calling exec. The child process calling “ls” is also a parent, so the same problem arises where there is no place to call wait(). Option 2 is the better implementation.