Gradey Cullins

HW1

2.13
One way to pass values to the OS is to place the values in registers that the OS can access. Another method is for the user process to push values onto its stack and for the OS to pop those values off the stack.

2.14
The simplest solution for profiling a C program is to use an existing tool like Valgrind. To achieve a desirable profile using this tool, I can place markers in my code that designate when to start and stop the instrumentation intervals in which the contained code will be measured.

I would compile the program with full debugger information i.e. using the -g flag, then run valgrind with the desired options and the compiled binary program as input.

To achieve a simple profiling of my code without an existing tool, I could add timing code around the regions of existing code that I want to profile. When the profiled region ends, I could print the elapsed time and the name of the completed segment.

3.9
First store all the process state, in the form of PCB, on it's corresponding kernel stack. Then choose a process from the ready queue and restore it's PCB state from its corresponding kernel stack. Part of this process must include loading the program counter so that the CPU can begin executing at the correct instruction of the new process.

3.12

8 processes.

3.17
Line X will have output: CHILD: 0 CHILD: -1 CHILD: -4 CHILD: -9 CHILD: -16
Line Y will have output: PARENT: 0 PARENT: 1 PARENT: 2 PARENT: 3 PARENT: 4