

# Written Assignment #1

Jakob Roberts - v00484900 - CSC 360

1. a) Kernel Mode:
  - complete and unrestricted access to system hardware
  - crashes halt entire PC
  - lowest level functions of OS that are most trusted
  - execute any instruction and reference any memory addressUser mode:
  - no direct access to hardware
  - crashes are always recoverable due to the protection provided by isolation
  - code running must conform to system API to interface with hardware or memory
- b) Both are needed to ensure safety of the system processes that are run by the kernel from the user processes/applications.
- c) **Context switch** is when the processor switches between a thread/process to another.  
**Mode switch** is when the CPU changes privilege levels.
- d) Pros:
  - stability
  - security
  - potentially more responsive
  - benefits for SMPCons:
  - Additional context switches may be required
  - slow inter-process communication could result in lowered performance

2. a)

0	or	0
2		1
1		2

b)

```
#include <stdio.h>
#include <unistd.h>

#define OUTPUT printf("%d\n", i)

main(){
    int i=0; OUTPUT;

    if(fork()){
        wait();
        i+=2; OUTPUT;
    } else {
        i++; OUTPUT; return(0);
    }
}
```

3.

- |    |                    |              |   |
|----|--------------------|--------------|---|
| a) | Running => Blocked | feasible     | some kind of I/O or event wait  |
| b) | Blocked => Running | not feasible | needs to return to the ready state first so the system scheduler can re-prioritize the process                |
| c) | Blocked => Ready   | feasible     | when there is an I/O or there is an even completion   |
| d) | Ready => Blocked   | not feasible | Has to be going from a running state. While in the ready state, no events causing a blocked state can happen. |
| e) | Ready => Running   | feasible     | the scheduler dispatch allows the process to run  |
| f) | Running => Ready   | feasible     | either a software or a hardware interrupt   |