CSC 360 W1

James Ryan – V00830984 – Jan 23, 2018

1.

a) User mode restricts things that the “user process” can access, such as memory, storage, etc. The kernel mode has very little, if not no restrictions, on access within the system.

b) It is needed to facilitate resource management and provide security so that a “user program” can’t access or make changes to resources it isn’t supposed to, such as memory not allocated to that program.

c) Context switch is used for switching the CPU between different process. Mode switch is used for switching the CPU between the user mode and the kernel mode

d) Pros: Smaller kernel size. Makes extending the OS easier. Better security as services run as a user process rather than a kernel process

Cons: Minimal process/memory management. Lower performance due to high overhead needed to run the system and its services

2.

a) Possible outputs:

0		0		0
1	or	2	or	1
2		1		

b)

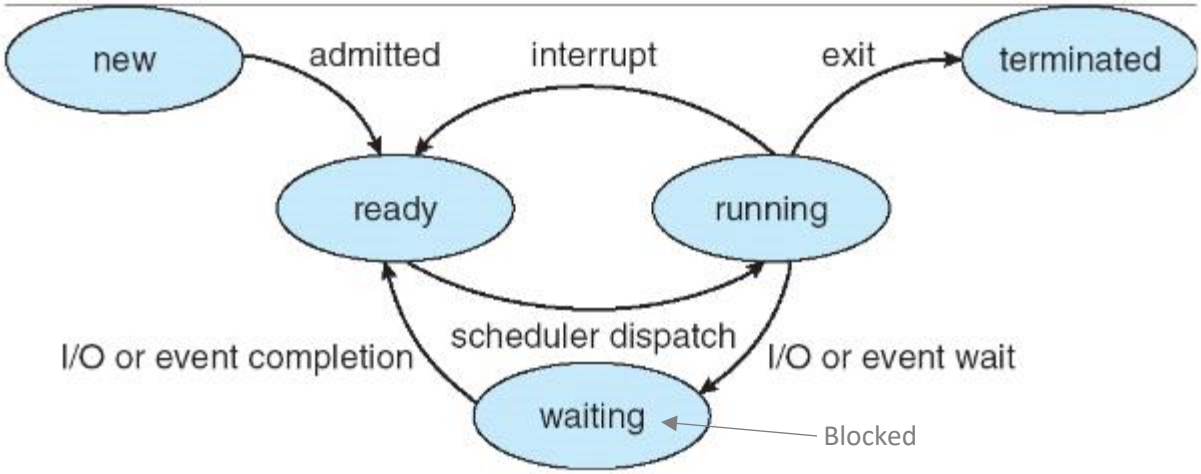
```
#define OUTPUT printf("%d\n", i)

main() {
  int i=0; OUTPUT;

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

wait(childProcessID);

3.



Screenshot from Lecture Slides on Processes

a) Running-to-blocked: Yes – for example, a process needs to read from secondary memory, which takes longer to access, so it switches to the blocked/waiting state while waiting for the read. This allows other processes currently in the ready state to run

b) Blocked-to-running: No – the process would need to be sent to the ready state to then be dispatched by the scheduler.

c) Blocked-to-ready: Yes – the secondary memory read that step a) was waiting on has finished, so the process moves to the ready state as it is ready to continue running.

d) Ready-to-blocked: No – there is no way for the program to trigger into the blocked/waiting state, because it is sitting idle while in the ready state and would need to be running to trigger the blocked/waiting state.

e) Ready-to-running: Yes – the system scheduler has triggered the process which was waiting in part c) to continue running.

f) Running-to-ready: Yes – A system interrupt has been made which results the process running from step e) to be paused and put back into the ready state so that the system can do what was required by the interrupt.