Jory Anderson

V00843894

CSC 360 Spring 2018

Written Assignment 1

1a)

- User applications are run in user mode

- System calls are run in kernel mode

- User applications needing to run system calls are transitioned to and from kernel mode using a system call interface and mode bit.

1b)

- Protects users from other users.

- Prevents the user application from writing over memory outside of the application space.

1c)

- Context Switching: Saving and restoring the execution state of a process from the PCB (Process Control Block)

- Mode Switching: The OS performing a system call (in kernel mode) on behalf of the user. Involves transitioning the application from user mode – to kernel mode – and back to user mode.

1d)

**Pros:**

- New services do need any intervention from the kernel.

- Smaller kernel, and therefore changes are few

- More fault-tolerant by design. The majority of the applications are running in user space, which means if they were to crash the OS could continue to run and possibly recover those services.

**Cons:**

- Pushing services out of the kernel creates more overhead for system functions, which decreases performance of the OS.

2a)

0

2

1

0

1

2

2b)

int main() {

int i=0; OUTPUT;

if (fork()) {

wait(NULL);

i+=2; OUTPUT;

} else {

i++; OUTPUT; return(0);

}

return 0;

}

3a)

Feasible. A running process could fork a child process, and wait for termination before continuing.

3b)

Not feasible. A process could be waiting on another to terminate. Once the second application is completed, the first process’s state is restored first before running again. It must be ready before being run again.

3c)

Feasible. Upon event completion a process’s execution state is restored from the state before the event occured. The process can then be resumed.

3d)

Not feasible. A process that is ‘ready’ isn’t necessarily running, so it cannot be blocked. The process would have to attempt to resume before being blocked.

3e)

Feasible. A process’s state is reloaded from its PCB, whose execution is resumed.

3f)

Feasible. An interrupt could tell a process to stop. The process is not waiting on anything, it is simply paused, or ‘hanging’.