Cameron Hartling
ID: 007735059

CSE 4600 Operating Systems

Homework Assignment 2: Processes

Part I Long Answer Questions

Question 1. (2 points) In the context of an operating system, what is a process and how does a process differ from a program? Additionally, please elaborate on the four main sections that make up the memory layout of a process.

In the context of an operating system a process is a program in execution. A program is an entity described in a static state while a process is the entity when the program is run and is stored in memory. The code section of the memory contains the program's instructions, the data section holds global and static variables, the heap contains dynamically allocated memory, and the stack holds temporary data when activating functions.

Question 2. (2 points) Please explain the various states a process can be in, and provide a detailed description of the transitions that occur between these states in an operating system.

The process states include new, ready, running, waiting, and terminated. The new state is where the process initializes and the program begins to execute. The ready state is where the process is waiting to be assigned to a CPU. The running state is where the process is executing the instructions. The waiting state is where the process is waiting for an event such as an I/O completion. The terminated state is where the process finishes executing. The transition between the new and ready state is where the process finishes initializing and is now ready to be assigned to a CPU. From the ready state the process gets assigned a CPU and begins to execute the instructions and moves into the running state. If the time slice expires while in the running state the process will move back to the ready state. If the process needs

an event to occur such as an I/O completion it will move to the waiting state and on completion of the required event will move back to the ready state. When the process is in the running state the process can be ended by a normal exit, an error, or it can be ended by another process. When this happens it will move into the terminated state.

Question 3. (2 points) Please provide detailed explanations for the following concepts: the degree of multiprogramming, context switch, process control block, scheduler, CPU-bound process, and I/O bound process.

Degree of Multiprogramming: The number of processes that are currently in memory.

Context Switch: A context switch is when a CPU changes between processes. To switch between processes the OS must save the state of the current process to its PCB and reload/restore the state of the new process from its PCB to registers.

Process Control Block (PCB): The PCB stores pieces of information associated with a specific process. It serves as a repository for all data needed to restart a process when it moves from the ready state to the running state.

Scheduler: The role of a scheduler is to select a process from the ready queue and allocate an open CPU core to the process.

CPU-Bound Process: A process that spends the majority of its time doing computations than it spends doing I/O.

I/O Bound Process: A process that spends the majority of its time doing I/O than it spends doing computations.

Question 4. (2 points) Explain the purpose and functionality of the fork() system call in operating systems. Describe how it creates a new child process and the differences between the parent and child processes after a successful fork() operation.

The fork() system call is used to create a new child process. The new process is created as a copy of the original process in the same address space as the original process allowing for the parent process to easily communicate with the child process. The parent and child processes have different process identifiers and the fork() of the parent process returns the pid of the child process while the fork() of the child process returns 0. The parent and child process will also have separate PCBs.

Question 5. (2 points) Please investigate and describe the process states and the process control block (struct proc) in xv6. How are the processes stored in xv6?

Question 6. (2 points) Please explain and compare two fundamental models of interprocess communication: shared memory and message passing in detail.

Shared memory is when a region of memory is shared among cooperating processes. These processes can communicate by reading and writing in the shared region and this can be used to share larger amounts of data. The major problem with this is the need for a mechanism to allow the processes to synchronize their actions when accessing the shared memory.

Message passing is when communication between processes are in the form of messages. This is used for smaller amounts of data and avoids the issues of having to synchronize. The processes use system calls to pass messages between each other.

Part II Programming Questions:

For the following 6 questions, please provide the screenshots of the program's output on a Linux OS.

Question 7. (3 points)

```
//Name the file as hw2_q7.cpp, compile the file using g++
```

```
I am process 92885
My parent is 92806
My x is 10
I am process 92886
My parent is 92885
My x is 10
```

Question 8. (3 points)

```
//Name the file as hw2_q8.cpp, compile the file using g++
```

```
I am the parent 92894
 my parent is 92806 and my x is 20
I am the child 92895
 my parent is 92894 and my x is 10
```

Question 9. (3 points)

```
//Name the file as hw2_q9.c, compile the file using gcc
```

```
I am parent 92904 and my child is 92905.
I am process 92904.
I am parent 92904 and I'm waiting for child 92905.
I am process 92905 and my parent is 92904.
I am process 92905.
I am process 92905 and I quit.
I am parent 92904 and my child 92905 has arrived. I quit too.
```

Question 10. (3 points)

```
//Name the file as hw2_q10.c, compile the file using g++, to run the executable, add 4 command line arguments and the last one should be a number ./q10 a1 a2 a3 11.1
```

```
5
./q10
a1
a2
a3
11.1
```

Question 11. (3 points)

```
//Name the file as hw2_q11.c, compile the file using g++,
```

```
[007735059@csusb.edu@jb358-5 hw2]$ g++ -o q11 hw2_q11.c
[007735059@csusb.edu@jb358-5 hw2]$ ./q11
I am parent 92940 and my child is 92941.
I am process 92940.
I am parent 92940 and I'm waiting for child 92941.
I am process 92941 and my parent is 92940.
I am process 92941.
total 124
-rw-r--r-- 1 007735059@csusb.edu domain users@csusb.edu   667 Sep 20 17:53 hw2_q10.c
-rw-r--r-- 1 007735059@csusb.edu domain users@csusb.edu  1087 Sep 20 17:53 hw2_q11.c
-rw-r--r-- 1 007735059@csusb.edu domain users@csusb.edu  1178 Sep 20 17:54 hw2_q12.c
-rw-r--r-- 1 007735059@csusb.edu domain users@csusb.edu   264 Sep 20 17:49 hw2_q7.cpp
-rw-r--r-- 1 007735059@csusb.edu domain users@csusb.edu   627 Sep 20 17:50 hw2_q8.cpp
-rw-r--r-- 1 007735059@csusb.edu domain users@csusb.edu  1045 Sep 20 17:50 hw2_q9.c
-rwxr-xr-x 1 007735059@csusb.edu domain users@csusb.edu 17608 Sep 20 18:01 q10
-rwxr-xr-x 1 007735059@csusb.edu domain users@csusb.edu 17864 Sep 20 18:03 q11
-rwxr-xr-x 1 007735059@csusb.edu domain users@csusb.edu 18320 Sep 20 17:58 q7
-rwxr-xr-x 1 007735059@csusb.edu domain users@csusb.edu 18320 Sep 20 18:00 q8
-rwxr-xr-x 1 007735059@csusb.edu domain users@csusb.edu 17808 Sep 20 18:00 q9
I am parent 92940 and my child 92941 has arrived.
```

Question 12. (3 points)

```
//Name the file as hw2_q12.c, compile the file using g++
//when running the executable q12, add command line arguments,
./q12 /bin/ls -l
```

```
[007735059@csusb.edu@jb358-5 hw2]$ g++ -o q12 hw2_q12.c
[007735059@csusb.edu@jb358-5 hw2]$ ./q12 /bin/ls -1
I am parent 92949 and my child is 92950.
I am process 92949.
I am parent 92949 and I'm waiting for child 92950.
I am process 92950 and my parent is 92949.
I am process 92950.
hw2_q10.c
hw2_q11.c
hw2_q12.c
hw2_q7.cpp
hw2_q8.cpp
hw2_q9.c
q10
q11
q12
q7
q8
q9
I am parent 92949 and my child 92950 has arrived.
```