**HW 1 -3600 Operating System**

**Submit your answers in Moodle as a word document or a pdf on or before (02/07/2020).**

* 1. **1.** Indicate whether each series of state transitions for a process is valid or invalid. Justify your answer if its invalid. **(3 points)**

a. new ➛ ready ➛ blocked ➛ ready

***Invalid, Ready >> Blocked, cannot be moved to block state without reaching running state first.***

* 1. b. running ➛ blocked ➛ ready ➛ blocked
  2. ***Invalid, Ready >> Blocked, cannot be moved to block state without reaching running state first.***
  3. c. new ➛ ready ➛ running ➛ ready
  4. ***Valid***

2. Match the following for a PCB of process P. **(6 points)**

|  |  |
| --- | --- |
| 1. CPU State | A) Every process is created by some other running process. If P is that running process. |
| 2. Process state | B) Contains information used by the scheduler to decide when p should run. The information typically records p's CPU time, the real time in the system, the priority, and any possible deadlines. |
| 3. Scheduling information | C) When p is stopped, the current state of the CPU, consisting of various hardware registers and flags, is saved in this field. The save information is copied back to the CPU when p resumes execution. |
| 4. Accounting Information | D) Records the identity of a child process in parent. |
| 5. Parent | E) Stores p's current state. Ex: Running, ready, or blocked. |
| 6. Children | F) Keeps track of information necessary for accounting and billing purposes. Ex: The amount of CPU time or memory used. |

3. From the list, indicate which PCB fields will not change during a **process's lifetime**. Why? (2 points)

Child, Parent, CPU\_state and process\_state

4. From the list, indicate which PCB fields may change while a process is in the **running state.** Why? (2 points)

Child, Parent, CPU\_state and process\_state

5. Learn the **top** command to display the resource utilization statistics of processes

. Open a terminal and type the **top** command.

. Start a browser and see the effect on the top display.

. Press Ctrl-Z to stop.

5.1 Observe and Write what did you notice? (2 points)

***The process for the browser started and jumped to the top of the list of processes. Ranging from 10-30% of the CPU usage, which was substantially more than the other processes.***

5.2 Compile any C program (Use a long loop - say while(1) to observe the effect) **in a new terminal** and observe the effect in top command in old terminal. Write what notice (2 points)

***The Program stayed consistently at the top of the process list, using around 85% CPU. Very cool to see how quickly it went back to normal after stopping the C program.***

5.3 From the top display, answer the following: – How much memory is free in the system? – Observe Which process is taking more CPU? – Which process has got maximum memory share? **(3 points)**

1. **13192.7**
2. **Firefox**
3. **Firefox**

5.4 Write a CPU bound C program (Program 1) and an I/O bound C program (program2 using more printf statements within while(1) loop), compile and execute both of them. Observe the effect of their CPU share using the top display and comment.

**Press Ctrl -Z to stop any running program. (5 points)**

**Program 1**

**#include <unistd.h>**

**#include <stdio.h>**

**int main(int argc, char \*argv[]) {**

unsigned int i,j;

while(1) {

j = 1;

for (i = 1; i <= 10; i++) {

j = j\*i;

}

}

**}**

**Program 2**

**#include <stdio.h>**

**#include <sys/time.h>**

**int main(int argc, char \*argv[]) {**

unsigned int i;

int count = 0;

struct timeval tv;

while(1) {

for(i = 0; i < 10; i++) {

gettimeofday(&tv, NULL);

**printf("%lu sec, %lu usec\n", tv.tv\_sec, tv.tv\_usec);**

}

count++;

**printf("round %d complete\n", count);**

}

**}**