

HW 1 -3600 Operating System

1. Indicate whether each series of state transitions for a process is valid or invalid. Justify your answer if its invalid. **(6 points)**
  - a. new → ready → blocked → ready
  - b. running → blocked → ready → blocked
  - c. new → ready → running → ready
2. 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
3. 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
4. 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.
- 4.1 Observe and Write what did you notice and what are some parameters you see there? **(5 points)**
- 4.2 Write the C program- [program2 given below using while \(1\) loop](#), let the top command run in the old terminal and now compile the program in **in a new terminal** and observe the parameters of top command, observe which process is taking more CPU? – Which process has got maximum memory share? Write your observations and justify. **Press Ctrl -Z to stop any running program.**  
**Include screenshots of terminal for full points (10 points)**
- 4.3 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 using two different terminals. Compare the effect of their CPU share using the top display in a third terminal and write your observation and justify.  
**Press Ctrl -Z to stop any running program. Include screenshots of terminal for full points (10 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);
    }
}
```