Tutorial 2 – Processes and Threads & Concurrency

Operating Systems CS 3SH3 Term 2, Winter 2022
Prof. Neerja Mhaskar

Tutorials are not mandatory. They are simply a tool for you understand the course concepts better.

Tutorial Format: The questions will be posted a day before or on the day of the tutorial on the course website. You can choose to solve these problems before hand and come in with your solutions. I or one of the TAs helping me will check your solutions. If you have all of the questions correct you can choose to leave. If you have any of them incorrect, it is recommended that you stay and understand how the problem is solved.

Solutions to the tutorial will not be posted online.

Q1) Using the program below explain what the output will be at LINE A and why?

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int value = 5;

int main()
{
pid_t pid;

  pid = fork();

  if (pid == 0) { /* child process */
    value += 15;
    return 0;
  }
  else if (pid > 0) { /* parent process */
    wait(NULL);
    printf("PARENT: value = %d",value); /* LINE A */
    return 0;
  }
}
```

Q2) Including the initial parent process, how many processes are created by the program shown below? Construct a tree of processes as explained in class for the processes created.

```
#include <stdio.h>
#include <unistd.h>


{
int main()
{
  int i;


{
  for (i = 0; i < 4; i++)
   fork();
{


  return 0;
}
```

Q4) Using the program below, identify the values of pid at lines A, B, C, and D. (Assume that the actual pids of the parent and child are 2600 and 2603, respectively.)

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int main()
{
pid_t pid, pid1;

  /* fork a child process */
  pid = fork();

  if (pid lt; 0) { /* error occurred */
    fprintf(stderr, "Fork Failed");
    return 1;
  }
  else if (pid == 0) { /* child process */
    pid1 = getpid();
    printf("child: pid = %d",pid); /* A */
    printf("child: pid1 = %d",pid1); /* B */
  }
  else { /* parent process */
    pid1 = getpid();
    printf("parent: pid = %d",pid); /* C */
    printf("parent: pid1 = %d",pid1); /* D */
    wait(NULL);
  }
  return 0;
}
```

Q3) When a process creates a new process using the `fork()` operation, which of the following states is shared between the parent process and the child process?
a. Stack
b. Heap
c. Shared memory segments

Q5) Using Amdahl's Law, calculate the speedup gain of an application that has a 60 percent parallel component for (a) two processing cores and (b) four processing cores.
Q6) Distinguish between parallelism and concurrency.
Q7) Distinguish between data and task parallelism.