PRASAD V POTLURI SIDDHARTHA INSTITUTE OF TECHNOLOGY(AUTONOMOUS), KANURU

DEPARTMENT OF CSE

Academic Year : 2025-26

Course Name : Operating Systems

Branch/Sem : B.Tech CSE II year II semester Assignment : 01

Registered Number:

Name of the student:

---

1. Analyse the following code and write the Output. (CO4 - L4)

```
#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;
   }
}
```

2. The Collatz conjecture concerns what happens when we take any positive integer *n* and apply the following algorithm:

$$n = \begin{cases} n/2, & \text{if n is even} \\ 3 \times n + 1, & \text{if n is odd} \end{cases}$$

The conjecture states that when this algorithm is continually applied, all positive integers will eventually reach 1. For example, if $n = 35$, the

sequence is

35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1

Write a C program using the fork() system call that generates this sequence in the child process. The starting number will be provided from the command line. For example, if 8 is passed as a parameter on the command line, the child process will output 8, 4, 2, 1. Because the parent and child processes have their own copies of the data, it will be necessary for the child to output the sequence. Have the parent invoke the wait() call to wait for the child process to complete before exiting the program. Perform necessary error checking to ensure that a positive integer is passed on the command line. (CO4- L4)

3. Write a multithreaded program that calculates various statistical values for a list of numbers. This program will be passed a series of numbers on the command line and will then create three separate worker threads. One thread will determine the average of the numbers, the second will determine the maximum value, and the third will determine the minimum value. For example, suppose your program is passed the integers (CO4- L4)

   90 81 78 95 79 72 85

   The program will report

   The average value is 82

   The minimum value is 72

   The maximum value is 95

   The variables representing the average, minimum, and maximum values will be stored globally. The worker threads will set these values, and the parent thread will output the values once the workers have exited. (We could obviously expand this program by creating additional threads that determine other statistical values, such as median and standard deviation.)

4. Consider the following set of processes, with the length of the CPU burst time given in milliseconds: (CO4- L4)

Process	Burst time	Priority
P1	2	2
P2	1	1
P3	8	4
P4	4	2
P5	5	3

   The processes are assumed to have arrived in the order *P*1, *P*2, *P*3, *P*4, *P*5, all at time 0.

   a. Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, nonpreemptive priority (a larger priority number implies a higher priority), and RR (quantum = 2).
   b. What is the turnaround time of each process for each of the scheduling algorithms in part a?
   c. What is the waiting time of each process for each of these scheduling algorithms?
   d. Which of the algorithms results in the minimum average waiting time (over all processes)?

5. The following processes are being scheduled using a preemptive, roundrobin scheduling algorithm. (CO4- L4)

Process	arrival	priority	Burst
P1	0	40	20
P2	25	30	25
P3	30	30	25
P4	60	35	15
P5	100	5	10
P6	105	10	10

Each process is assigned a numerical priority,with a higher number indicating a higher relative priority. In addition to the processes listed below, the system also has an **idle task** (which consumes no CPU resources and is identified as *Pidle*). This task has priority 0 and is scheduled whenever
the system has no other available processes to run. The length of a time quantum is 10 units. If a process is preempted by a higher-priority process, the preempted process is placed at the end of the queue.

a. Show the scheduling order of the processes using a Gantt chart.

b. What is the turnaround time for each process?

c. What is the waiting time for each process?

d. What is the CPU utilization rate?