Processes and CPU Scheduling

Overview

The practise questions are designed to complement and enhance your knowledge of topics covered in the lectures. Not all answers will be readily found on the lecture notes and slides, and you may be required to engage in some self-learning to complete the questions.

Suggested answers to the practise questions shall be provided at a later date. It is strongly advised that you attempt these questions on your own, and discuss them with your peers before consulting the suggested answers.

Optional questions are designed for further challenge. For these questions, answers may or may not be provided.

Practise Questions

1. Briefly describe main differences between a process and a program.

2. Describe the actions taken by a kernel to context-switch between processes.

3. What are the two models of inter-process communication (IPC)? What are the strengths and weakness of the two approaches?

4. In the program code shown in Figure 1, what will be the output of `LINE A`? Explain your answer.

```c
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.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;
    }
}
```

Figure 1

5. Including the initial parent process, how many processes are created by the program shown in Figure 2?

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

int main() {

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

    /* fork another child process */
    fork();

    /* fork yet another child process */
    fork();

    return 0;
}
```

Figure 2

6. Consider the following set of processes in Table I, with the length of the CPU burst time given in milliseconds

Table I

Process	Burst Time (ms)
P1	30
P2	27
P3	35
P4	29
P5	21

Draw the Gantt chart for round-robin scheduling and compute the average waiting time and turnaround time for a time quantum of 29 milliseconds. How many context switches are required?

7. Repeat Question 6 with the following set of processes shown in Table II and a time quantum of 10ms.

Table II

Process	Burst Time (ms)
P1	10
P2	15
P3	20
P4	26
P5	32

Optional Questions

8. The Fibonacci sequence is the sequence of numbers that starts with 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

   Formally, it can be expressed as

$$\begin{aligned} u_0 &= 0 \\ u_1 &= 1 \\ u_n &= u_{n-1} + u_{n-2} \end{aligned}$$

   Write a C program using the `fork()` system call that generates the Fibonacci sequence in the child process. The number of the sequence shall be provided in the command line (e.g., through a command line parameter).

Example: If 5 is provided, the first five numbers in the Fibonacci sequence will be output by the child process, i.e., 0, 1, 1, 2, 3.

Hint: 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. You may also want to perform necessary error checking to ensure that a non-negative number is passed through the command line.

END OF DOCUMENT