

Exercises on Process Management in C

Operating Systems Module

University of Bouira - Computer Science Department
Academic Year 2024-2025

Exercise 5

Write a C program that creates a hierarchy of processes using the `fork()` function. The program should:

- Create a parent process.
- From the parent, create two child processes.
- Each child process should create one grandchild process.

Question: Draw the process tree resulting from this program.

Exercise 6

Consider the following expression:

$$z := ((g + h) \times (i - j)) + \left(\frac{k}{l} + (m - n)\right)$$

- **Tasks:** Decompose this expression into independent tasks $T_1, T_2, T_3, \dots, T_n$.
- **Question:** Use Bernstein's conditions to determine which tasks can be executed in parallel.
- **Programming Task:** Implement a C program using `fork()` to evaluate this expression in parallel where possible.

Bernstein's Conditions for Parallelism

To determine which tasks can be parallelized, we use Bernstein's Conditions. These conditions state that two tasks T_i and T_j can execute in parallel if:

1. They do not have overlapping input variables.
2. They do not have overlapping output variables.
3. The output of one task is not used as input by the other.

Task Decomposition and Analysis: Break down the expression into tasks and apply Bernstein's conditions to identify which tasks can run in parallel.

Exercise 7

Write a C program that creates a specified number of child processes (e.g., 3) and has each child print its PID and that of its parent. The program should:

- Use a loop to create each child.
- Ensure that each child executes a code block where it prints its PID and parent PID.
- Ensure that the parent waits for all children to complete before terminating.