**Practical File: System Calls in C**

**Objective:** This practical aims to demonstrate the use of various system calls in C, specifically ‘fork()’, 'getpid()', 'getppid()', 'wait()', and 'exit()'. These system calls are essential for process management in a UNIX/Linux environment.

**System Calls Overview:**

* fork(): Creates a new process (child process) which is a copy of the parent process.
* getpid(): Returns the process ID of the calling process.
* getppid(): Returns the process ID of the parent of the calling process.
* wait(): Makes the parent process wait until the child process completes execution.
* exit(): Terminates the process and returns a status code to the parent process.

**I. Creating a Child Process Using 'fork()'**

Program 1: Creating a Child Process

This program demonstrates the creation of a child process using 'fork()'.

Program1.c

**II. Printing Process IDs**

Program 2: Printing Process IDs of Parent and Child

This program demonstrates how to print the process IDs of both the parent and child processes.

Program2.c

**III. Listing Files Using a Child Process and Parent Waiting**

Program 3: Child Process Listing Files, Parent Waiting

This program demonstrates the use of 'fork()', 'wait()', and 'exit()' where the child process lists files in the directory, and the parent process waits for the child to finish.

Program3.c

**IV. What Happens If the Parent Dies Before the Child?**

Program 4: Parent Process Terminating Before Child

This program illustrates what happens when the parent process dies before the child process. A new child process is also created to observe the behavior.

Program4.c