

EXP 1: Create a new process by invoking the appropriate system call. Get the process identifier of the currently running process and its respective parent using system calls and display the same using a C program.

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

int main() {
    pid_t pid;

    // Create a new process
    pid = fork();

    if (pid < 0) {
        // Error in process creation
        perror("fork failed");
        return 1;
    } else if (pid == 0) {
        // Child process
        printf("Child Process:\n");
        printf(" PID: %d\n", getpid());
        printf(" Parent PID: %d\n", getppid());
    } else {
        // Parent process
        printf("Parent Process:\n");
        printf(" PID: %d\n", getpid());
        printf(" Parent PID: %d\n", getppid());
    }
    return 0;
}
```

Sample Output

Parent Process:

PID: 1378

Parent PID: 1371

Child Process:

PID: 1379

Parent PID: 1378

=== Code Execution Successful ===