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.

   Program:

[*] system call 1.cpp

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

int main(void) {
  pid_t pid = fork();

  if(pid == 0) {
    printf("Child => PPID: %d PID: %d\n", getppid(), getpid());
    exit(EXIT_SUCCESS);
  }
  else if(pid > 0) {
    printf("Parent => PID: %d\n", getpid());
    printf("Waiting for child process to finish.\n");
    wait(NULL);
    printf("Child process finished.\n");
  }
  else {
    printf("Unable to create child process.\n");
  }

  return EXIT_SUCCESS;
}
```

Output:

```
Parent => PID: 5254
Waiting for child process to finish.
Child => PPID: 5254 PID: 5255
Child process finished.
```