System call Implementation

vfork() is used when the child process will call exec() or _exit() immediately.

It's more efficient than fork() in such cases because it avoids copying the parent’s memory space.

Always use _exit() in the child to prevent issues from shared memory space with the parent.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
int main() {
    pid_t pid;
    printf("Before vfork\n");
    pid = vfork();  // vfork is used to create a new process without copying the address space
    if (pid < 0) {
        perror("vfork failed");
        exit(1);
    } else if (pid == 0) {
        // Child process
        printf("Child process (PID: %d)\n", getpid());
        execlp("/bin/ls", "ls", "-l", NULL);  // Replace child with 'ls -l' command
        perror("execlp failed");
        _exit(1);  // Use _exit instead of exit in child process after vfork
    } else {
        // Parent process
        printf("Parent process (PID: %d), child PID: %d\n", getpid(), pid);
    }

    return 0;
}
```