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

int main() {
    // parent prints things before child

    // point of choosing which process to execute
    int pid = fork();
    if (pid == 0) {
        printf("I am the child: %d\n", getpid());
        printf("child is searching for something
         special\n");
        int result = execl("/usr/sbin/grep", "grep", "-n",
         "special", "simple.c", 0);
        if (-1 == result) {
            perror("failed to run program");
            return result;
        }
        // we never return ...
    } else if (pid > 0) {
        printf("I am the parent: %d\n", getpid());

        printf("parent is %d and waiting for child %d\n",
         getpid(), pid);
        int status;
        int result = wait(&status);
        if (-1 == result) {
            // failed case
            perror("could not wait for child process");
        } else {
            printf("child status code: %d\n", status);
            if (0 != status)
                printf("child failed to do its task\n");
            else
                printf("child was successful!\n");
        }
    } else {
        // fail case
        perror("failed to fork");
    }

    return 0;
}
```