分析以下代码中使用的linux系统调用。要求使用strace工具

```c
#include<stdio.h>
#include<stdlib.h>
#include<errno.h>

int split(char* cmd, char output[5][256]){
    int count = 0; //yongyu jisuan canshu
    int length = strlen(cmd);
    int i,j=0,flag=0;
    for(i=0; i<length; i++){
        if(cmd[i] == ' '){
            if(count >0 && j!=0){
                output[count-1][j] = '\0';
            }
            flag = 0;
        }else{
            if(flag == 0){
                count++;
                j = 0;
                flag = 1;
            }
            output[count -1][j] = cmd[i];
            j++;
        }

    }
    return count;
}
char cmd[256];
char params[5][256];
int counter=0;
void main(){
    int ret;
    printf("Hello World, this is Linux!");
    while(1){
        printf(">");
        fgets(cmd,256,stdin);
        if (cmd[strlen(cmd) - 1] == '\n')
        {
                cmd[strlen(cmd) - 1] = '\0';
            }
        if(fork() == 0)
        {
            counter = split(cmd, params);
                if(counter != 0)
            {
                switch(counter)
                {
                    case 0: break;
                    case 1:
                    {
                        execlp(params[0], params[0], NULL);
                    }
                    break;
```

```c
            case 2:
            {
                execlp(params[0], params[0] , params[1], NULL);
            }
            break;
            case 3:
            {
                printf("%s %s %s\n" ,params[0] ,params[1], params[2]);
                execlp(params[0], params[0], params[1], params[2],
                    (char*)0);
            }
            break;
            default:
            {
                printf("Illegal Input!\n");
            }
        } // end switch
    }
        perror(cmd);
        exit(errno);
    }else{
        wait(&ret);
        printf("child process return %d\n",ret);
    }
  }
    return 0;
}
```

输入命令：strace -f -F -o ./straceout.txt ./test

使用strace工具运行后得到的代码为:

```
execve("./test", ["./test"], [/* 36 vars */]) = 0
brk(0)                    = 0x8556000
access("/etc/ld.so.nohwcap", F_OK)    = -1 ENOENT (No such file or directory)
mmap2(NULL, 8192, PROT_READIPROT_WRITE, MAP_PRIVATEIMAP_ANONYMOUS, -1, 0) =
0xb7727000
access("/etc/ld.so.preload", R_OK)    = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLYIO_CLOEXEC) = 3
fstat64(3, {st_mode=S_IFREGI0644, st_size=62136, ...}) = 0
mmap2(NULL, 62136, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7717000
close(3)                  = 0
access("/etc/ld.so.nohwcap", F_OK)    = -1 ENOENT (No such file or directory)
open("/lib/i386-linux-gnu/libc.so.6", O_RDONLYIO_CLOEXEC) = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0@\226\1\0004\0\0\0"..., 512) = 512
fstat64(3, {st_mode=S_IFREGI0755, st_size=1734120, ...}) = 0
mmap2(NULL, 1747676, PROT_READIPROT_EXEC, MAP_PRIVATEIMAP_DENYWRITE, 3, 0) =
0xb756c000
mprotect(0xb7710000, 4096, PROT_NONE)   = 0
mmap2(0xb7711000, 12288, PROT_READIPROT_WRITE, MAP_PRIVATEIMAP_FIXEDI
MAP_DENYWRITE, 3, 0x1a4) = 0xb7711000
mmap2(0xb7714000, 10972, PROT_READIPROT_WRITE, MAP_PRIVATEIMAP_FIXEDI
MAP_ANONYMOUS, -1, 0) = 0xb7714000
close(3)                  = 0
```

mmap2(NULL, 4096, PROT_READIPROT_WRITE, MAP_PRIVATEIMAP_ANONYMOUS, -1, 0) = 0xb756b000
set_thread_area({entry_number:-1 -> 6, base_addr:0xb756b900, limit:1048575, seg_32bit:1, contents:0, read_exec_only:0, limit_in_pages:1, seg_not_present:0, useable:1}) = 0
mprotect(0xb7711000, 8192, PROT_READ)   = 0
mprotect(0x8049000, 4096, PROT_READ)   = 0
mprotect(0xb774a000, 4096, PROT_READ)   = 0
munmap(0xb7717000, 62136)   = 0
fstat64(1, {st_mode=S_IFCHRl0620, st_rdev=makedev(136, 0), ...}) = 0
mmap2(NULL, 4096, PROT_READIPROT_WRITE, MAP_PRIVATEIMAP_ANONYMOUS, -1, 0) = 0xb7726000
write(1, "Hello world, this is Linux!\n", 28) = 28
fstat64(0, {st_mode=S_IFCHRl0620, st_rdev=makedev(136, 0), ...}) = 0
mmap2(NULL, 4096, PROT_READIPROT_WRITE, MAP_PRIVATEIMAP_ANONYMOUS, -1, 0) = 0xb7725000
write(1, ">", 1)   = 1
read(0, "Orangeloo\n", 1024)   = 10
clone(child_stack=0, flags=CLONE_CHILD_CLEARTIDICLONE_CHILD_SETTIDISIGCHLD, child_tidptr=0xb756b968) = 2779
wait4(-1, [{WIFEXITED(s) && WEXITSTATUS(s) == 2}], 0, NULL) = 2779
--- SIGCHLD (Child exited) @ 0 (0) ---
write(1, "child process return 512 \n", 26) = 26
write(1, ">", 1)   = 1
read(0, "exit\n", 1024)   = 5
clone(child_stack=0, flags=CLONE_CHILD_CLEARTIDICLONE_CHILD_SETTIDISIGCHLD, child_tidptr=0xb756b968) = 2780
wait4(-1, [{WIFEXITED(s) && WEXITSTATUS(s) == 2}], 0, NULL) = 2780
--- SIGCHLD (Child exited) @ 0 (0) ---
write(1, "child process return 512 \n", 26) = 26
write(1, ">", 1)   = 1
read(0, 0xb7725000, 1024)   = ? ERESTARTSYS (To be restarted)
--- SIGHUP (Hangup) @ 0 (0) ---
+++ killed by SIGHUP +++

系统调用:
        execve、write、read、clone、wait4...