

```

#include <iostream>
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <ctype.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdlib.h>
using namespace std ;

#define MAX_CHILD_NUMBER 3
#define SLEEP_INTERVAL 2

int proc_number = 0;
void do_something(int num);

int main(int argc, char** argv) {

    int child_proc_number = MAX_CHILD_NUMBER;
    int ch;
    //每个进程的进程ID
    pid_t pid[10] = {0};

    if(argc > 1) {
        //命令行参数中的第一个参数表示建立几个子进程，最多10个
        child_proc_number = atoi(argv[1]);
        child_proc_number = (child_proc_number > 10)?10:child_proc_number;
    }

    for(int i=0; i<child_proc_number; i++){

        if((pid[i] = fork()) == 0) {
            //保存子进程id
            proc_number=i;
            do_something(proc_number);
            break;
        }
        else if(pid[i] < 0) {
            break;
        }
        else {
            cout << "产生的子进程ID" << pid[i] << endl;
        }
        //这里填写代码，建立child_proc_number个进程
        //子进程要执行
        //父进程子进程的id保存到pid[i]
    }
    cout << "父进程执行!" << endl;
    //让用户选择杀死那个进程，输入数字表示杀死该进程
    //输入q退出
    int status;
    cin >> ch;
    while(ch != -1) {
        if(waitpid(pid[ch], &status, WNOHANG) == 0) {

```

```
int ret = kill(pid[ch], SIGKILL);
if(ret) {
    cout << "杀死进程失败!" << endl;
}
else {
    cout << "成功杀死进程!" << endl;
}
}
else {
    cout << strerror(errno) << endl;
}
}
cin >> ch;
}
//这里填写杀死本组的所有进程
return 0;
}

void do_something(int num) {
    for(;;) {
        printf("this is process No. %d running!\n", num+3, num);
        sleep(2);
    }
}
```