同济大学计算机网络

实验报告

姓名：涂远鹏-1652262

四.如何杀死守护进程？如何杀死守护进程的子进程？
1.在 04 子目录下写 test4-1.c，循环 10 次，每个 3 秒产生一个子进程.....写配套的 makefile，make 后生成 test4-1 可执行文件：
答：
根据目标功能编写的 test4-1.c 如下：

```
#include <stdio.h>
#include <sys/types.h>
#include <signal.h>
#include <string.h>
#include <sys/wait.h>

int main()
{
    int i,j;
    pid_t pid1;
    pid1=fork();
    if(pid1==-1)
    return 0;
    if(pid1==0)
    {
      for(i=0;i<10;)
      {
        pid_t pid = fork();
        if (pid == 0)
        {
          for(;;)
          {
            printf("%d  %d  1652262  sub\n",(int)getppid(),(int)getpid());
            sleep(15);
          }
       }
        i++;
        sleep(3);
    }
    while(1)
    {
        printf("%d  %d  1652262  main\n",(int)getppid(),(int)getpid());
        sleep(5);
    }
    return 0;
    }
    if(pid1>0)
    return 0;

}
```

编写的 makefile 文件如下：

```
test:test4-1 test4-2
$(test):
        cc -o
clean:
        rm test4-?
~
```

执行 make 操作，并运行生成的 test4-1 可执行文件并将输出的打印信息到 output1 文件，进程的生成如下：

```
[root@RHEL74-SVR 04]# ./test4-1 >output1 &
[1] 2342
[root@RHEL74-SVR 04]# ps -a -l
F S   UID    PID   PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
1 S     0   2343      1  0  80   0 -  1052 hrtime pts/0    00:00:00 test4-1
1 S     0   2344   2343  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2345   2343  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
0 R     0   2346   2285  0  80   0 - 38293 -      pts/0    00:00:00 ps
[1]+  完成                  ./test4-1 > output1
[root@RHEL74-SVR 04]# ps -a -l
F S   UID    PID   PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
1 S     0   2343      1  0  80   0 -  1052 hrtime pts/0    00:00:00 test4-1
1 S     0   2344   2343  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2345   2343  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2347   2343  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2348   2343  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
0 R     0   2349   2285  0  80   0 - 38293 -      pts/0    00:00:00 ps
[root@RHEL74-SVR 04]# ps -a -l
F S   UID    PID   PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
1 S     0   2343      1  0  80   0 -  1052 hrtime pts/0    00:00:00 test4-1
1 S     0   2344   2343  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2345   2343  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2347   2343  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2348   2343  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2350   2343  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2351   2343  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2352   2343  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2353   2343  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
0 R     0   2354   2285  0  80   0 - 38293 -      pts/0    00:00:00 ps
[root@RHEL74-SVR 04]# ps -a -l
```

2.如何杀死 test4-1 分裂出来的一个子进程？

答：

通过 kill+子进程号杀死子进程，此处杀死的是 2343 子进程：

```
[root@RHEL74-SVR 04]# kill 2343
[root@RHEL74-SVR 04]# ps -a -l
F S   UID    PID   PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
1 S     0   2344      1  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2345      1  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2347      1  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2348      1  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2350      1  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2351      1  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2352      1  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2353      1  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2355      1  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
1 S     0   2356      1  0  80   0 -  1053 hrtime pts/0    00:00:00 test4-1
0 R     0   2359   2285  0  80   0 - 38293 -      pts/0    00:00:00 ps
[root@RHEL74-SVR 04]#
```

3.如何快速杀死 test4-1 分裂出来的全部子进程？

答：

(1)首先在 test4-1.c 中加入 func_waitpid 函数：

```c
#include <stdio.h>
#include <sys/types.h>
#include <signal.h>
#include <string.h>
#include <sys/wait.h>

void func_waitpid(int signo) {
    pid_t pid;
    int stat;
    while( (pid = waitpid(-1, &stat, WNOHANG)) > 0 ) ;
    return;
}


int main()
{
    signal(SIGCHLD, &func_waitpid);
    int i,j;
    pid_t pid1;
    pid1=fork();
    if(pid1==-1)
    return 0;
    if(pid1==0)
    {
      for(i=0;i<10;)
      {
        pid_t pid = fork();
        if (pid == 0)
        {
          for(;;)
          {
            printf("%d  %d  1652262  sub\n",(int)getppid(),(int)getpid());
            sleep(15);
          }
       }
        i++;
        sleep(3);
    }
    while(1)
    {
        printf("%d  %d  1652262  main\n",(int)getppid(),(int)getpid());
        sleep(5);
    }
    return 0;
    }
    if(pid1>0)
    return 0;

}
```

(2)执行 test4-1 可执行文件,执行 For p_pid in `ps -ef|grep test4-1|egrep -v grep|awk '{print $2}'`杀死所有 test4-1 的子进程：

```
[root@RHEL74-SVR 04]# ps -a -l
F S   UID    PID   PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
1 S     0   2436      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2437   2436  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2438   2436  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2439   2436  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2440   2436  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2441   2436  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2442   2436  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2443   2436  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2445   2436  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2446   2436  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2448   2436  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
0 R     0   2449   2394  0  80   0 - 38293 -      pts/1    00:00:00 ps
[root@RHEL74-SVR 04]# for p_pid in `ps -ef|grep test4-1|egrep -v grep|awk '{print $2}'`
> do
> pkill -9 -P $p_pid
> done
[root@RHEL74-SVR 04]# ps -a -l
F S   UID    PID   PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
1 S     0   2436      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
0 R     0   2475   2394  0  80   0 - 38293 -      pts/1    00:00:00 ps
[root@RHEL74-SVR 04]#
```

4.如何杀死 test4-1，其子进程会发生哪些变化？

答：杀死身为父进程的 test4-1，可以观察到子进程正常运行并未发生退出但是由 test4-1 产生的子进程的 PPID(父进程标识)全部变为 1，即 init 进程：

```
[root@RHEL74-SVR 04]# ./test4-1 >output1 &
[1] 2477
[root@RHEL74-SVR 04]# ps -a -l
F S   UID    PID   PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
1 S     0   2436      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2478      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2479   2478  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2480   2478  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2481   2478  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2482   2478  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2483   2478  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2484   2478  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2485   2478  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2486   2478  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2487   2478  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2488   2478  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
0 R     0   2496   2394  0  80   0 - 38293 -      pts/1    00:00:00 ps
[1]+  完成                  ./test4-1 > output1
[root@RHEL74-SVR 04]# kill 2478
[root@RHEL74-SVR 04]# ps -a -l
F S   UID    PID   PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
1 S     0   2436      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2479      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2480      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2481      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2482      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2483      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2484      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2485      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2486      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2487      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2488      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
0 R     0   2510   2394  0  80   0 - 38293 -      pts/1    00:00:00 ps
[root@RHEL74-SVR 04]#
```

5.写 test4-2.c，要求与 test4-1 相同，但是要求杀死 test4-2 后，它的全部子进程自动退出：

答：

(1)编写 test4-2.c，在 test4-1.c 的基础上，在子进程生成中加入信号控制函数 ptctrl：

```c
#include <stdio.h>
#include <sys/types.h>
#include <signal.h>
#include <string.h>
#include <sys/wait.h>
#include <stdlib.h>
#include <sys/prctl.h>
void func_waitpid(int signo) {
    pid_t pid;
    int stat;
    while( (pid = waitpid(-1, &stat, WNOHANG)) > 0 ) ;
    return;
}
int main()
{
    signal(SIGCHLD, &func_waitpid);
    int i,j;
    pid_t pid1;
    pid1=fork();
    if(pid1==-1)
    return 0;
    if(pid1==0)
    {
        for(i=0;i<10;)
        {
          pid_t pid = fork();
          if (pid == 0)
          {
            while(1)
            {
              prctl(PR_SET_PDEATHSIG, SIGHUP);
              printf("%d  %d  1652262  sub\n",(int)getppid(),(int)getpid());
              sleep(15);
            }
          exit(0);
        }
        i++;
        sleep(3);
    }
    while(1)
    {
        printf("%d  %d  1652262  main\n",(int)getppid(),(int)getpid());
        sleep(5);
    }
    exit(0);
    system("killall test4-2");
}
    if(pid1>0)
    return 0;

}
```

(2)运行生成的 test4-2，并杀死其父进程，可以看到父进程生成的子进程全部被杀，说明测试成功：

```
[root@RHEL74-SVR 04]# ./test4-2 >output2 &
[1] 2521
[root@RHEL74-SVR 04]# ps -a -l
F S   UID    PID   PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
1 S     0   2436      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2479      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2480      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2481      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2482      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2483      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2484      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2485      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2486      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2487      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2488      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2522      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-2
1 S     0   2523   2522  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-2
1 S     0   2524   2522  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-2
1 S     0   2525   2522  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-2
1 S     0   2526   2522  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-2
1 S     0   2527   2522  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-2
1 S     0   2528   2522  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-2
1 S     0   2529   2522  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-2
1 S     0   2530   2522  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-2
1 S     0   2531   2522  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-2
1 S     0   2532   2522  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-2
0 R     0   2533   2394  0  80   0 - 38293 -      pts/1    00:00:00 ps
[1]+  完成                  ./test4-2 > output2
[root@RHEL74-SVR 04]# kill 2522
[root@RHEL74-SVR 04]# ps -a -l
F S   UID    PID   PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
1 S     0   2436      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2479      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2480      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2481      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2482      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2483      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2484      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2485      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2486      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2487      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
1 S     0   2488      1  0  80   0 -  1053 hrtime pts/1    00:00:00 test4-1
0 R     0   2534   2394  0  80   0 - 38293 -      pts/1    00:00:00 ps
[root@RHEL74-SVR 04]#
```