同济大学计算机网络

实验报告

姓名：涂远鹏-1652262

题目：捕获 kill-9 信号的示例

方法一：

1. 由于 SIGKILL 信号无法被捕捉和忽略，并且受到 SIGKILL 信号的进程会马上被 KILL 掉，所以可以从发送信号端进行判断，首先根据发送 SIGKILL 信号的 pid 设置一个信号随时监控该 pid 的存在，倘若发出信号后该 pid 在 ps -ef 进程列表中消失同时判断 kill(pid_create,SIGKILL)的返回值，如果返回值为-1 则接收 kill -9 信号失败，如果大于 0 则接收信号成功，并判断该 pid 进程是否存在，如若不存在则进一步说明进程收到 SIGKILL 成功：

```
if((fp = popen(command,"r")) == NULL)
{
    printf("(fp = popen(command,\"r\")) == NULL\n");
}

if( (fgets(buf,BUFSZ,fp))!= NULL )
{
    count = atoi(buf);

    if((count - 1) == 0)
    {

        printf("已收到kill -9信号\n");

        //可以用system()开启test
    }

    else
    {
        printf("未收到kill -9信号\n");
    }

}
pclose(fp);
fp=NULL;
```

```
if(ret==-1)
{
   find_pid();
   printf("signal SIGKILL kill failed\n");
}
else {
printf("signal SIGKILL kill received\n");
find_pid();
}
```

2.测试结果如下，显示”signal SIGKILL kill received”前后 test-extra-2 被杀死，显示 test-extra-2 接收 SIGKILL 信号成功：

```
[root@RHEL74-SVR 03]# ./test-extra-2
[root@RHEL74-SVR 03]# ./test-extra-1
[root@RHEL74-SVR 03]#
OUCH1! - I got signal 10
ps
   PID TTY          TIME CMD
  2181 pts/0    00:00:00 bash
  2440 pts/0    00:00:00 test-extra-2
  2444 pts/0    00:00:00 test-extra-1
  2451 pts/0    00:00:00 ps
[root@RHEL74-SVR 03]# ps
   PID TTY          TIME CMD
  2181 pts/0    00:00:00 bash
  2440 pts/0    00:00:00 test-extra-2
  2444 pts/0    00:00:00 test-extra-1
  2452 pts/0    00:00:00 ps
[root@RHEL74-SVR 03]# signal SIGKILL kill received
已收到kill -9信号
ps
   PID TTY          TIME CMD
  2181 pts/0    00:00:00 bash
  2444 pts/0    00:00:00 test-extra-1
  2456 pts/0    00:00:00 ps
[root@RHEL74-SVR 03]# ps
   PID TTY          TIME CMD
  2181 pts/0    00:00:00 bash
  2444 pts/0    00:00:00 test-extra-1
  2475 pts/0    00:00:00 ps
```

方法二：

1.另外也可以添加 strace 对 test-extra-2 进程进行跟踪，指令为 strace -e trace=write -s 200 -f -p +进程 pid,在 test-extra-1 中用 system()函数执行该指令，测试结果如下，显示进程是被 SIGKILL 杀死的：

```
[root@RHEL74-SVR 03]# ./test-extra-2
[root@RHEL74-SVR 03]# ./test-extra-1
PID: 2806

[root@RHEL74-SVR 03]#
OUCH1! - I got signal 10
strace -e trace=write -s 200 -f -p 2806
strace: Process 2806 attached
signal SIGKILL kill received
+++ killed by SIGKILL +++
[root@RHEL74-SVR 03]# 已收到kill -9信号
```

重新进行测试，可以看到运行 test-extra-2 之后该进程的 pid 号为 2335，使用 strace -e trace=write -s 200 -f -p 2335 进行监控，然后运行 test-extra-1 杀死 test-extra-2 进程，strace 结果显示 killed by SIGKILL,说明进程接收到了 SIGKILL 信号:

```
[root@RHEL74-SVR 03]# ./test-extra-2
[root@RHEL74-SVR 03]# ./test-extra-1
PID: 2335

[root@RHEL74-SVR 03]#
OUCH1! - I got signal 10
signal SIGKILL kill received
已收到kill -9信号
```

```
root       2308      1  0 18:47 ?        00:00:00 /usr/sbin/abrt-dbus -t133
root       2335      1  0 18:48 pts/0    00:00:00 ./test-extra-2
root       2336   1450  8 18:48 ?        00:00:00 sshd: root@pts/1
root       2340   2336  1 18:48 pts/1    00:00:00 -bash
root       2375   2340  0 18:48 pts/1    00:00:00 ps -ef
[root@RHEL74-SVR ~]# strace -e trace=write -s 200 -f -p 2335
strace: Process 2335 attached
--- SIGUSR1 {si_signo=SIGUSR1, si_code=SI_USER, si_pid=2384, si_uid=0} ---
write(1, "\n", 1)                       = 1
write(1, "OUCH1! - I got signal 10\n", 25) = 25
strace: Process 2388 attached
strace: Process 2389 attached
strace: Process 2390 attached
[pid  2389] write(1, "   PID TTY          TIME CMD\n  2384 pts/0    00:00:00 test-extra-1\n", 67) = 67
[pid  2390] write(1, "2\n", 2)          = 2
[pid  2389] +++ exited with 0 +++
[pid  2390] +++ exited with 0 +++
[pid  2388] --- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=2389, si_uid=0, si_status=0, si_
utime=0, si_stime=1} ---
[pid  2388] +++ exited with 0 +++
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=2388, si_uid=0, si_status=0, si_utime=0, si_
stime=0} ---
+++ killed by SIGKILL +++
[root@RHEL74-SVR ~]#
```