《Linux应用程序开发实验报告》

题目: 进程控制

班级: 17计算机4班

学号: 10417426

姓名: 沈雨晨

东南大学成贤学院计算机系

2020年4月10日

1. **实验目的**

掌握进程的概念

掌握进程的创建和管理

掌握守护进程的概念、建立、启动及调试方法

1. **实验报告要求：**

1、以下是对execlp进行调试之后显示的调试信息，根据此信息回答问题

执行父进程的进程号是多少？

执行子进程的进程号是多少？

[Detaching after fork from child process 14692. (Try `set detach-on-fork off'.)]

UID       PID         PPID     C         STIME     TTY       TIME          CMD

root     14562      1           0         22:12       ?          00:00:00     gnome-terminal

root     14565     14562     0         22:12      ?           00:00:00     gnome-pty-helper

root     14566     14562     0         22:12     pts/1      00:00:00      bash

root     14675     14566     0         22:15     pts/1      00:00:00      gdb execlp

root     14676     14675     0         22:15     pts/1      00:00:00      /root/7/execlp

root     14692     14676     0         22:15     pts/1       00:00:00      ps -ef

辅助说明：gnome-terminal就是终端进程，bash是shell命令的版本类型，多数Linux发行版本采用bash作为其命令版本；

 答; 执行父进程的进程号是：14676；执行子进程的进程号是：14692

2、使用gdb调试waitpid()，如果首先执行的是父进程，会出现什么情况？

      如果想要调试子进程应该如何做？？？

 答：如果首先执行父进程，父进程未阻塞，不在意子进程是否退出状态，waitpid()返回值为0，检测到子进程未退出,睡1s,然后继续判断waitpid()返回值，为子进程号，然后捕获子进程退出，结束程序执行；如果想要调试子进程，将“ waitpid(pc, NULL, WNOHANG);”变为“waitpid(pc,NULL,0);”

3、已知multi\_proc可执行程序的一个可能的执行结果，如下所示：

[root@localhost ~]# ./multi\_proc

In child1: execute 'ls -l'

In child2: sleep for 5 seconds and then exit

In father process:

总计 96

 -rw------- 1 root root 1155 2014-09-26 anaconda-ks.cfg

drwxr-xr-x 2 root root 4096 2014-09-26 Desktop

-rw-r--r-- 1 root root 35507 2014-09-26 install.log

-rw-r--r-- 1 root root 4399 2014-09-26 install.log.syslog

drwxrwxrwx 10 root root 4096 02-26 08:27 linux

-rwxr-xr-x 1 root root 5760 03-16 12:42 multi\_proc

drwxr-xr-x 3 root root 4096 01-05 08:57 vmtools

Get child1 exit code

The child2 process has not exited!

The child2 process has not exited!

The child2 process has not exited!

The child2 process has not exited!

The child2 process has not exited!

Get child2 exit code

[root@localhost ~]#

假设CPU是单核，即某时刻只有一个进程在CPU上运行，试说明这些结果是如何运行出来的？

答：首先创建子进程1，然后在子进程1未出错的情况下打印输出“In child1: execute 'ls -l'”，会调用execlp()函数，然后创建子进程2，在子进程2未出错的情况下打印输出“In child2: sleep for 5 seconds and then exit"，睡5秒后退出，然后父进程中，执行两个waitpid函数，先打印输出“In father process:”然后阻塞父进程等待子进程1调用execlp()打印输出list列表后即“总计 96

 -rw------- 1 root root 1155 2014-09-26 anaconda-ks.cfg

drwxr-xr-x 2 root root 4096 2014-09-26 Desktop

-rw-r--r-- 1 root root 35507 2014-09-26 install.log

-rw-r--r-- 1 root root 4399 2014-09-26 install.log.syslog

drwxrwxrwx 10 root root 4096 02-26 08:27 linux

-rwxr-xr-x 1 root root 5760 03-16 12:42 multi\_proc

drwxr-xr-x 3 root root 4096 01-05 08:57 vmtools”

子进程1结束输出“Get child1 exit code”后在进行后面操作，此时不可以一边执行父进程其他语句一边等待子进程的消息，然后父进程对子进程2则是非阻塞式等待，可以一边执行父进程其他语句一边等待子进程的消息，每隔一秒打印输出“The child2 process has not exited!”，子进程2结束输出“Get child2 exit code”，至此程序运行结束，父进程也退出。

两个子进程执行顺序无序，抢占式执行。

是否可以设置进程间的执行顺序，如果可以应该怎么做？

可以，在父进程中加入wait(),当有一次子进程在执行的时候，父进程进入等待状态，执行子进程直到完成，然后接着判断是否还有别的子进程需要执行，有则父进程继续等待，执行子进程2.

 4、针对GDB多进程调试视频中的问题。

为什么在GDB中，以默认的调试方式调试waitpid程序；将子进程sleep(5);上下加两条打印语句，如下所示：

printf("The child process is running!\n"); /\*子进程暂停5s\*/

sleep(5);

printf("The child process is quitted!\n"); /\*子进程正常退出\*/

exit(0);

两条printf()之间父进程只打印4条The child process has not exited语句；如下面代码所示：

(gdb) r

Starting program: /root/7/waitpid\_GDB多进程调试 \_子进程有打印语句

Error in re-setting breakpoint 1: Function "waitpid\_GDB多进程调试" not defined.

Detaching after fork from child process 32028.

The child process has not exited

The child process is running!

The child process has not exited

The child process has not exited

The child process has not exited

The child process has not exited

The child process is quitted!

The father process quit because the child process quit.

The return of waitpid() is the child process PID : 32028

Program exited normally.

(gdb)

而在./waitpid\_GDB多进程调试Shell命令行执行中两条printf()之间父进程却打印5条The child process has not exited语句？

[tom@localhost ~]$ ./waitpid\_GDB多进程调试

The child process is running!

The child process has not exited

The child process has not exited

The child process has not exited

The child process has not exited

The child process has not exited

The child process is quitted!

The father process quit because the child process quit.

The return of waitpid() is the child process PID : 7059

因为运行的环境不一样，不同的父子进程执行顺序不一样，gdb调试的时候系统调试父进程并且父进程不阻塞，而子进程不受影响继续执行，所以先进父进程，此时子进程也未退出，所以先打印输出The child process has not exited，然后父进程休眠1s,此时子进程打印The child process is running，然后子进程休眠5s后会退出,这期间父进程每隔1s继续执行打印输出语句The child process has not exited，5s一到，子进程退出不再打印。

5、守护进程创建的步骤是什么？它有什么作用？

创建守护进程的步骤：

1.创建子进程，父进程退出

2.为子进程创建新会话并担任组长

3.改变当前目录为根目录

4.重设文件权限掩码

5.关闭文件描述符

作用：为系统运行提供了重要的安全保护

1. 实验心得

本次实验加深了対进程概念的理解以及守护进程的概念，建立和调试方法。

对于进程编程：

fork()函数作用是从已有的进程中创建一个新进程

返回值不是零，表示当前进程在父进程当中

exec函数族就提供了一个在进程中启动另一个程序执行的方法

其中：

execlp把当前进程打印出来 ps -ef

execle把由指针数组指定的两个字符串添加到当前子进程的环境变量

execve在指定路径下执行env命令并且将其添加到当前子进程的环境变量

对于exit（）和\_exit（）函数，都是用来终止进程的。

exit() 带缓存区，\_exit()直接使进程停止，并清除使用的内存空间。

对于waitpid()等待pid，一个进程等待另外一个进程

wait()是用于使父进程阻塞，直到一个子进程结束或者该进程接到一个指定的信号为止