

实验报告

实验（六）

题 目 TinyShell

微壳

专 业 计算机类

学 号 1160300524

班 级 1603005

学 生 冯运

指导教师 郑贵滨

实验地点 G712

实验日期 12月15日

计算机科学与技术学院

目 录

第1章 实验基本信息...	- 3 -
1.1 实验目的...	- 3 -
1.2 实验环境与工具...	- 3 -
1.2.1 硬件环境...	- 3 -
1.2.2 软件环境...	- 3 -
1.2.3 开发工具...	- 3 -
1.3 实验预习...	- 3 -
第2章 实验预习...	- 4 -
2.1 进程的概念、创建和回收方法（5分）...	- 4 -
2.2 信号的机制、种类（5分）...	- 4 -
2.3 信号的发送方法、阻塞方法、处理程序的设置方法（5分）...	- 4 -
2.4 什么是 shell，功能和处理流程（5分）...	- 4 -
第3章 TinyShell 测试...	- 5 -
3.1 TinyShell 设计...	- 5 -
第4章 总结...	- 5 -
4.1 请总结本次实验的收获...	- 5 -
4.2 请给出对本次实验内容的建议...	- 5 -
参考文献...	- 7 -

第 1 章 实验基本信息

1.1 实验目的

理解现代计算机系统进程与并发的基本知识

掌握 linux 异常控制流和信号机制的基本原理和相关系统函数

掌握 shell 的基本原理和实现方法

深入理解 Linux 信号响应可能导致的并发冲突及解决方法

培养 Linux 下的软件系统开发与测试能力

1.2 实验环境与工具

1.2.1 硬件环境

X64 CPU; 2GHz; 2G RAM; 256GHD Disk 以上

Windows7 64 位以上; VirtualBox/Vmware 11 以上; Ubuntu 16.04 LTS
64 位/优麒麟 64 位

1.2.2 软件环境

§ Windows7 64 位以上; VirtualBox/Vmware 11 以上;
Ubuntu 16.04 LTS 64 位/优麒麟 64 位

1.2.3 开发工具

VirtualBox/Vmware 11 以上; Ubuntu 16.04 LTS 64 位/优麒麟 64 位

1.3 实验预习

见第二章

第 2 章 实验预习

总分 20 分

2.1 进程的概念、创建和回收方法（5 分）

进程是一个执行中的程序实例，他表示的是一个程序正常运行的各种状态的一个集合，包括存放在内存中的代码和数据，栈，寄存器和 PC 等等程序的状态。

每个进程都可以创建自己的进程，并且可以在这个进程的上下文中运行自己的代码和或其他应用程序。

回收方法：当一个进程处于终止的状态时，会在系统中占用一定的资源，直到被父进程回收，并从系统中完全的清除。

2.2 信号的机制、种类（5 分）

信号主要用于内核通知进程发生的一些异常，进程接收到异常信号后，进行一些异常处理。

信号由硬件底层产生的信号，内核发送给进程的信号和进程发送给别的进程（包括自己的进程）几种。

2.3 信号的发送方法、阻塞方法、处理程序的设置方法（5 分）

发送方法：内核通过更新目的进程的上下文的某个状态，发送一个信号给目的进程。

- 1、程序中可以使用 `Kill()` 函数发送信号给别的进程或者给自己发送信号
- 2、使用 `/bin/kill` 可以向另外的进程发送任意的信号
- 3、从键盘发送信号

4、用 alarm 函数发送信号

阻塞方法:

隐式阻塞: 当一个处理程序正在处理信号 `s`, 这是又有一个 `s` 传送过来, 将会变成待处理信号被阻塞

显式阻塞: 程序可以使用 `sigpromask()` 函数和其他函数处理。通过修改阻塞信号集合中的元素来控制阻塞的过程。

2.4 什么是 shell, 功能和处理流程 (5 分)

Shell 是一个交互型应用级程序, 他代表用户运行其他程序。

功能:

Shell 通过执行一系列的读/求值步骤, 然后终止。通过读取用户输入的命令行, 解析命令行参数, 最后按照参数执行任务/

处理流程: 读取命令行, 调用 `parseline` 解析命令行参数, 如果是前台的任务, `shell` 会一直等待直到当前的进程结束, 如果是后台的进程, `shell` 可以创一个单独的进程运行, 不影响前台进程的进行。

■
■
■

第3章 TinyShell 的设计与实现

总分 45 分

3.1 设计

3.1.1 void eval(char *cmdline)函数 (10 分)

函数功能：使用 `parseline` 函数解析出来的参数，然后调用 `builtin_cmd` 函数，用于判断是否是内部指令，如果不是内部指令，`eval()` 函数就创建一个子进程执行所请求的程序，如果是后台程序，就直接进行新的循环，等待下一个命令行，如果是前台应用，它会一直等待子进程运行结束，并将其回收。

参数：char *cmdline 这个参数是用户输入的命令行字符串

处理流程：先调用 `parseline()` 函数，将用户输入的命令行解析成几个参数，然后调用 `builtin_cmd()` 函数判断，然后调用 `builtin_cmd` 函数，用于判断是否是内部指令，如果不是内部指令，`eval()` 函数就创建一个子进程执行所请求的程序，如果是后台程序，就直接进行新的循环，等待下一个命令行，如果是前台应用，它会一直等待子进程运行结束，并将其回收。

要点分析：

这个函数是用于处理命令行的主要函数，主要的地方在于新建一个子进程来运行目的程序，然后负责判断该进程的运行状态，最后回收该进程，结束这次运行。

3.1.2 int builtin_cmd(char **argv)函数 (5 分)

函数功能：负责判断是否是 shell 的内部命令，如果是内部命令，就直接执行，如果不是内部指令，就返回 0，让 `eval()` 函数去处理。

参数：char **argv 这是一个二元数组，是命令行解析出来的各种参数。

处理流程：先根据 `argv[0]` (命令行解析出来的第一个参数) 判断是否是内部参数，如果是，就直接指令这个命令，如果不是，就 `return 0`，交给 `eval()` 函数去处理。

要点分析：

这个函数主要就是几个 if 条件判断，使用 `strcmp()` 来判断 `argv[0]` 是否和 `fg`, `bg`, `quit`, `jobs` 这些内部的命令相同，如果是，就调用内部相应的函数去执行。

3.1.3 void do_bgfg(char **argv) 函数 (5 分)

函数功能：根据传入的参数，如果是 `fg`，就将某个指定的任务设置为前台运行，如果是 `bg`，就将指定的任务设定为后台运行。

参数：char **argv 是解析后的命令行的参数

处理流程：先判断是否输入了任务号，再判断这个任务号是否存在，之后判断是 **bg** 还是 **fg**，如果是 **bg**，就将任务的状态设置为 **BG**，如果是 **fg**，就将任务的状态设置为 **FG**。

要点分析：

需要将各种异常情况判断清楚，并作出相应的处理，不能有遗漏，否则就会出现运行的错误。

3.1.4 void waitfg(pid_t pid) 函数（5 分）

函数功能：将 **shell** 的运行暂停，直到前台没有要运行的进程

参 数：pid 这个是在正在运行的前台进程号

处理流程：设置一个循环，不断的检测前台是否有正在运行的进程，如果有进程正在运行，就不断循环，否则就结束返回。

要点分析：

需要判断是否有正在运行的进程，控制好循环进行的条件，也可以再循环之间设置休眠，减小程序的计算负荷。

3.1.5 void sigchld_handler(int sig) 函数（10 分）

函数功能：当有进程暂停或者终止，就执行这个函数，将终止的进程回收，并从任务组中删除记录。如果是有进程停止，就在任务组中将相应的任务状态设置为 **ST**，表示这个进程处于暂停的状态。

参 数：sig 是内核发送的异常信号。

处理流程：先调用 `waitpid(-1,&status,WNOHANG|WUNTRACED)` 函数，接收导致返回的进程号，使用 `WIFSTOPPED(status)` 来针对子进程自己给自己发送信号，而父进程无法判断子进程是由于什么信号结束的进行判断，如果是真，表示这个进程自己发送信号给自己的进程停止，要将这个进程的状态设置为 **ST**，否则判断 `WIFSIGNALED(status)`，如果为真，表示这个进程自己发送终止信号给自己，需要将这个进程从任务组中删除，否则再判断这个进程由外部原因结束的情况，作出相应的处理。

要点分析：

要点在于判断子进程自己发送信号给自己进程的特殊情况，这种情况 **shell** 无法获取子进程是由何种信号产生异常，需要特殊判断处理。

3.2 程序实现（tsh.c 的全部内容）（10 分）

重点检查代码风格：

（1） 用较好的代码注释说明——5 分

(2) 检查每个系统调用的返回值——5分

代码见附件

第 4 章 TinyShell 测试

总分 15 分

4.1 测试方法

针对 tsh 和参考 shell 程序 tshref，完成测试项目 4.1-4.15 的对比测试，并将测试结果截图或者通过重定向保存到文本文件（例如：`./sdriver.pl -t trace01.txt -s ./tsh -a "-p" > tshresult01.txt`）。

4.2 测试结果评价

tsh 与 tshref 的输出在一下两个方面可以不同：

- (1) PID
 - (2) 测试文件 `trace11.txt`, `trace12.txt` 和 `trace13.txt` 中的 `/bin/ps` 命令，每次运行的输出都会不同，但每个 `mysplit` 进程的运行状态应该相同。
- 除了上述两方面允许的差异，tsh 与 tshref 的输出相同则判为正确如不同则给出原因分析。

4.3 自测试结果

4.3.1 测试用例 trace01.txt 的输出截图（1 分）

tsh 测试结果		tshref 测试结果	
测试结论	相同		

4.3.2 测试用例 trace02.txt 的输出截图（1 分）

tshref 测试结果	
	
tsh 测试结果	
测试结论	相同

4.3.3 测试用例 trace03.txt 的输出截图（1 分）

tsh 测试结果		tshref 测试结果	
<div>终端</div> 			
测试结论	相同		

4.3.4 测试用例 trace04.txt 的输出截图（1分）

tshref 测试结果	
 <pre> 1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab\$ make test04 ./sdriver.pl -t trace04.txt -s ./tsh -a "-p" # # trace04.txt - Run a background job. # tsh> ./myspin 1 & [1] (4597) ./myspin 1 & 1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab\$ make rtest04 ./sdriver.pl -t trace04.txt -s ./tshref -a "-p" # # trace04.txt - Run a background job. # tsh> ./myspin 1 & [1] (4603) ./myspin 1 & 1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab\$ </pre>	
tsh 测试结果	
测试结论	相同

4.3.5 测试用例 trace05.txt 的输出截图（1分）

tsh 测试结果	tshref 测试结果
 <pre> 1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab\$ make test05 ./sdriver.pl -t trace05.txt -s ./tsh -a "-p" # # trace05.txt - Process jobs builtin command. # tsh> ./myspin 2 & [1] (4636) ./myspin 2 & tsh> ./myspin 3 & [2] (4638) ./myspin 3 & tsh> jobs [1] (4636) Running ./myspin 2 & [2] (4638) Running ./myspin 3 & 1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab\$ make rtest05 ./sdriver.pl -t trace05.txt -s ./tshref -a "-p" # # trace05.txt - Process jobs builtin command. # tsh> ./myspin 2 & [1] (4645) ./myspin 2 & tsh> ./myspin 3 & [2] (4647) ./myspin 3 & tsh> jobs [1] (4645) Running ./myspin 2 & [2] (4647) Running ./myspin 3 & 1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab\$ </pre>	

测试结论	相同
------	----

4.3.6 测试用例 trace06.txt 的输出截图（1分）

tsh 测试结果	tshref 测试结果
----------	-------------

```

终端
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$ make test06
./sdriver.pl -t trace06.txt -s ./tsh -a "-p"
#
# trace06.txt - Forward SIGINT to foreground job.
#
tsh> ./myspin 4
Job [1] (5642) terminated by signal 2
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$ make rtest06
./sdriver.pl -t trace06.txt -s ./tshref -a "-p"
#
# trace06.txt - Forward SIGINT to foreground job.
#
tsh> ./myspin 4
Job [1] (5648) terminated by signal 2
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$

```

测试结论	相同
------	----

4.3.7 测试用例 trace07.txt 的输出截图（1分）

tsh 测试结果	tshref 测试结果
----------	-------------

```

终端
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$ make test07
./sdriver.pl -t trace07.txt -s ./tsh -a "-p"
#
# trace07.txt - Forward SIGINT only to foreground job.
#
tsh> ./myspin 4 &
[1] (5690) ./myspin 4 &
tsh> ./myspin 5
Job [2] (5692) terminated by signal 2
tsh> jobs
[1] (5690) Running ./myspin 4 &
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$ make rtest07
./sdriver.pl -t trace07.txt -s ./tshref -a "-p"
#
# trace07.txt - Forward SIGINT only to foreground job.
#
tsh> ./myspin 4 &
[1] (5699) ./myspin 4 &
tsh> ./myspin 5
Job [2] (5701) terminated by signal 2
tsh> jobs
[1] (5699) Running ./myspin 4 &
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$

```


测试结论	相同

4.3.8 测试用例 trace08.txt 的输出截图（1 分）

tsh 测试结果	tshref 测试结果
----------	-------------

```

终端
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$ make test08
./sdriver.pl -t trace08.txt -s ./tsh -a "-p"
#
# trace08.txt - Forward SIGTSTP only to foreground job.
#
tsh> ./myspin 4 &
[1] (10030) ./myspin 4 &
tsh> ./myspin 5
Job [2] (10032) stopped by signal 20
tsh> jobs
[1] (10030) Running ./myspin 4 &
[2] (10032) Stopped ./myspin 5
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$ make rtest08
./sdriver.pl -t trace08.txt -s ./tshref -a "-p"
#
# trace08.txt - Forward SIGTSTP only to foreground job.
#
tsh> ./myspin 4 &
[1] (10039) ./myspin 4 &
tsh> ./myspin 5
Job [2] (10041) stopped by signal 20
tsh> jobs
[1] (10039) Running ./myspin 4 &
[2] (10041) Stopped ./myspin 5
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$

```

测试结论	相同

4.3.9 测试用例 trace09.txt 的输出截图（1 分）

tsh 测试结果	tshref 测试结果
 <pre> 终端 1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab\$ make test09 ./sdriver.pl -t trace09.txt -s ./tsh -a "-p" # # trace09.txt - Process bg builtin command # tsh> ./myspin 4 & [1] (10168) ./myspin 4 & tsh> ./myspin 5 Job [2] (10170) stopped by signal 20 tsh> jobs [1] (10168) Running ./myspin 4 & [2] (10170) Stopped ./myspin 5 tsh> bg %2 [2] (10170) ./myspin 5 tsh> jobs [1] (10168) Running ./myspin 4 & [2] (10170) Running ./myspin 5 1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab\$ make rtest09 ./sdriver.pl -t trace09.txt -s ./tshref -a "-p" # # trace09.txt - Process bg builtin command # tsh> ./myspin 4 & [1] (10179) ./myspin 4 & tsh> ./myspin 5 Job [2] (10181) stopped by signal 20 tsh> jobs [1] (10179) Running ./myspin 4 & [2] (10181) Stopped ./myspin 5 tsh> bg %2 [2] (10181) ./myspin 5 tsh> jobs [1] (10179) Running ./myspin 4 & [2] (10181) Running ./myspin 5 1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab\$ </pre>	
测试结论	相同

4.3.10 测试用例 trace10.txt 的输出截图（1 分）

tsh 测试结果	tshref 测试结果
 <pre> 终端 1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab\$ make test10 ./sdriver.pl -t trace10.txt -s ./tsh -a "-p" # # trace10.txt - Process fg builtin command. # tsh> ./myspin 4 & [1] (10239) ./myspin 4 & tsh> fg %1 Job [1] (10239) stopped by signal 20 tsh> jobs [1] (10239) Stopped ./myspin 4 & tsh> fg %1 tsh> jobs 1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab\$ make rtest10 ./sdriver.pl -t trace10.txt -s ./tshref -a "-p" # # trace10.txt - Process fg builtin command. # tsh> ./myspin 4 & [1] (10249) ./myspin 4 & tsh> fg %1 Job [1] (10249) stopped by signal 20 tsh> jobs [1] (10249) Stopped ./myspin 4 & tsh> fg %1 tsh> jobs 1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab\$ </pre>	

测试结论	相同

4.3.11 测试用例 trace11.txt 的输出截图（1分）

tsh 测试结果	tshref 测试结果
----------	-------------

```

1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$ make test11
./sdriver.pl -t trace11.txt -s ./tsh -a "-p"
#
# trace11.txt - Forward SIGINT to every process in foreground process group
#
tsh> ./mysplit 4
Job [1] (10340) terminated by signal 2
tsh> /bin/ps a
  PID TTY          STAT       TIME COMMAND
  963  tty7      Ssl+      6:53  /usr/lib/xorg/Xorg -core :0 -seat seat0 -auth /var/run/lightdm/root/:0 -nolisten tcp vt7 -novtswitch
  965  tty1      Ss+       0:00  /sbin/agetty --noclear tty1 linux
 4223 pts/4    Ss        0:00  bash
 4734 pts/4    R         48:20  ./tsh -p
 9685 pts/4    R         9:06  ./tsh -p
 9687 pts/4    T         0:00  ./mystop 2
 9777 pts/4    R         5:56  ./tsh -p
 9953 pts/4    R         4:10  ./tsh -p
 9955 pts/4    T         0:00  ./mystop 2
10335 pts/4    S+        0:00  make test11
10336 pts/4    S+        0:00  /bin/sh -c ./sdriver.pl -t trace11.txt -s ./tsh -a "-p"
10337 pts/4    S+        0:00  /usr/bin/perl ./sdriver.pl -t trace11.txt -s ./tsh -a -p
10338 pts/4    R+        0:00  ./tsh -p
10341 pts/4    S         0:00  ./mysplit 4
10343 pts/4    R         0:00  /bin/ps a
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$ make rtest11
./sdriver.pl -t trace11.txt -s ./tshref -a "-p"
#
# trace11.txt - Forward SIGINT to every process in foreground process group
#
tsh> ./mysplit 4
Job [1] (10353) terminated by signal 2
tsh> /bin/ps a
  PID TTY          STAT       TIME COMMAND
  963  tty7      Ssl+      6:53  /usr/lib/xorg/Xorg -core :0 -seat seat0 -auth /var/run/lightdm/root/:0 -nolisten tcp vt7 -novtswitch
  965  tty1      Ss+       0:00  /sbin/agetty --noclear tty1 linux
 4223 pts/4    Ss        0:00  bash
 4734 pts/4    R         48:23  ./tsh -p
 9685 pts/4    R         9:10  ./tsh -p
 9687 pts/4    T         0:00  ./mystop 2
 9777 pts/4    R         5:59  ./tsh -p
 9953 pts/4    R         4:13  ./tsh -p
 9955 pts/4    T         0:00  ./mystop 2
10348 pts/4    S+        0:00  make rtest11
10349 pts/4    S+        0:00  /bin/sh -c ./sdriver.pl -t trace11.txt -s ./tshref -a "-p"
10350 pts/4    S+        0:00  /usr/bin/perl ./sdriver.pl -t trace11.txt -s ./tshref -a -p
10351 pts/4    S+        0:00  ./tshref -p
10356 pts/4    R         0:00  /bin/ps a
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$

```

测试结论	相同

4.3.12 测试用例 trace12.txt 的输出截图（1 分）

tsh 测试结果

tshref 测试结果

```

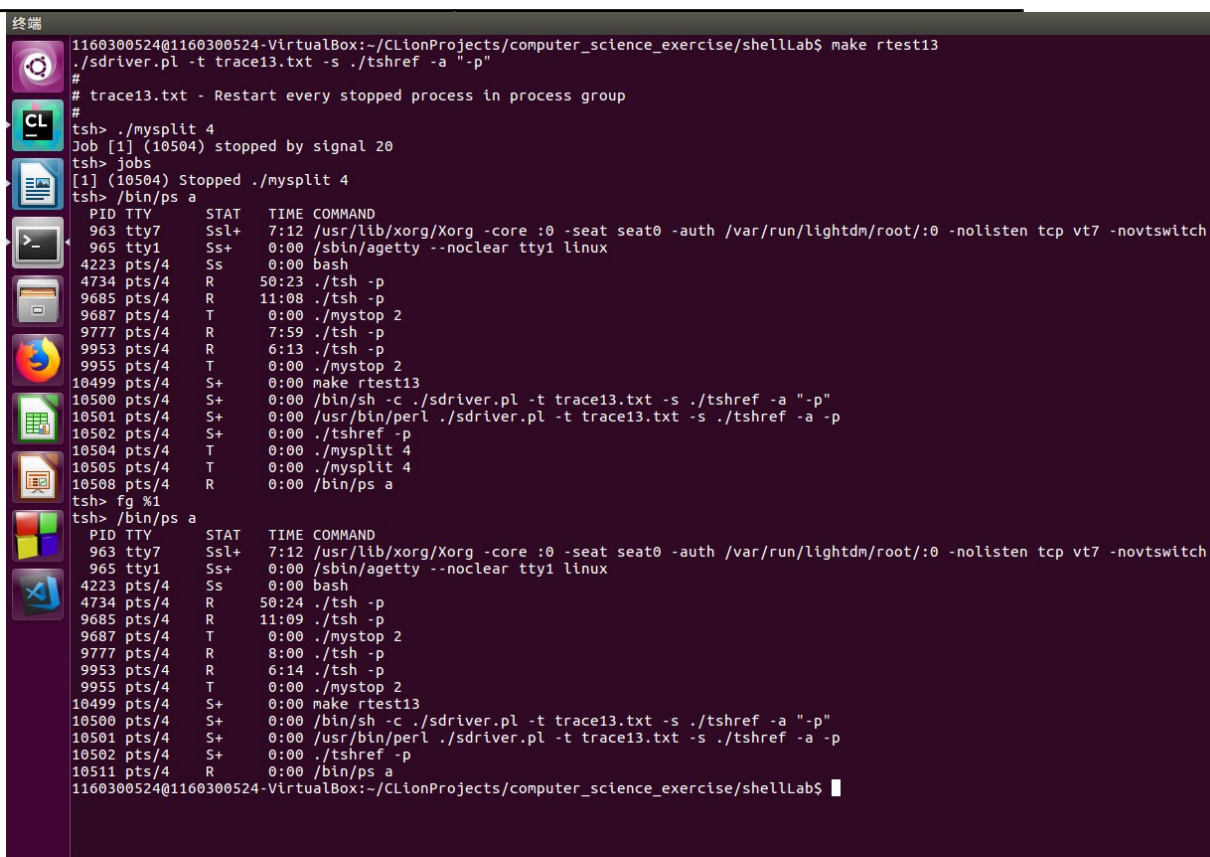
终端
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$ make test12
./sdriver.pl -t trace12.txt -s ./tsh -a "-p"
#
# trace12.txt - Forward SIGTSTP to every process in foreground process group
#
tsh> ./mysplit 4
Job [1] (10403) stopped by signal 20
tsh> jobs
[1] (10403) Stopped ./mysplit 4
tsh> /bin/ps a
  PID TTY          STAT       TIME COMMAND
   963 tty7        Ssl+      6:58 /usr/lib/xorg/Xorg -core :0 -seat seat0 -auth /var/run/lightdm/root/:0 -nolisten tcp vt7 -novtswitch
   965 tty1        Ss+       0:00 /sbin/agetty --noclear tty1 linux
  4223 pts/4      Ss        0:00 bash
  4734 pts/4      R         48:59 ./tsh -p
  9685 pts/4      R         9:45 ./tsh -p
  9687 pts/4      T         0:00 ./mystop 2
  9777 pts/4      R         6:34 ./tsh -p
  9953 pts/4      R         4:48 ./tsh -p
  9955 pts/4      T         0:00 ./mystop 2
 10398 pts/4      S+        0:00 make test12
 10399 pts/4      S+        0:00 /bin/sh -c ./sdriver.pl -t trace12.txt -s ./tsh -a "-p"
 10400 pts/4      S+        0:00 /usr/bin/perl ./sdriver.pl -t trace12.txt -s ./tsh -a -p
 10401 pts/4      R+        0:00 ./tsh -p
 10403 pts/4      T         0:00 ./mysplit 4
 10404 pts/4      S         0:00 ./mysplit 4
 10407 pts/4      R         0:00 /bin/ps a
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$ make rtest12
./sdriver.pl -t trace12.txt -s ./tshref -a "-p"
#
# trace12.txt - Forward SIGTSTP to every process in foreground process group
#
tsh> ./mysplit 4
Job [1] (10413) stopped by signal 20
tsh> jobs
[1] (10413) Stopped ./mysplit 4
tsh> /bin/ps a
  PID TTY          STAT       TIME COMMAND
   963 tty7        Ssl+      6:58 /usr/lib/xorg/Xorg -core :0 -seat seat0 -auth /var/run/lightdm/root/:0 -nolisten tcp vt7 -novtswitch
   965 tty1        Ss+       0:00 /sbin/agetty --noclear tty1 linux
  4223 pts/4      Ss        0:00 bash
  4734 pts/4      R         49:03 ./tsh -p
  9685 pts/4      R         9:49 ./tsh -p
  9687 pts/4      T         0:00 ./mystop 2
  9777 pts/4      R         6:38 ./tsh -p
  9953 pts/4      R         4:52 ./tsh -p
  9955 pts/4      T         0:00 ./mystop 2
 10408 pts/4      S+        0:00 make rtest12
 10409 pts/4      S+        0:00 /bin/sh -c ./sdriver.pl -t trace12.txt -s ./tshref -a "-p"
 10410 pts/4      S+        0:00 /usr/bin/perl ./sdriver.pl -t trace12.txt -s ./tshref -a -p
 10411 pts/4      S+        0:00 ./tshref -p
 10413 pts/4      T         0:00 ./mysplit 4
 10414 pts/4      T         0:00 ./mysplit 4
 10417 pts/4      R         0:00 /bin/ps a
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$ 

```

测试结论 相同

4.3.13 测试用例 trace13.txt 的输出截图（1 分）

tsh 测试结果	tshref 测试结果
	



测试结论	相同
------	----

4.3.14 测试用例 trace14.txt 的输出截图 (1 分)

tsh 测试结果	tshref 测试结果
----------	-------------


```
终端
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$ make
./sdriver.pl -t trace14.txt -s ./tsh -a "-p"
#
# trace14.txt - Simple error handling
#
tsh> ./bogus
./bogus: Command not found
tsh> ./myspin 4 &
[1] (10599) ./myspin 4 &
tsh> fg
fg command requires PID or %jobid argument
tsh> bg
bg command requires PID or %jobid argument
tsh> fg a
fg: argument must be a PID or %jobid
tsh> bg a
bg: argument must be a PID or %jobid
tsh> fg 9999999
(9999999): No such process
tsh> bg 9999999
(9999999): No such process
tsh> fg %2
%2: No such job
tsh> fg %1
Job [1] (10599) stopped by signal 20
tsh> bg %2
%2: No such job
tsh> bg %1
[1] (10599) ./myspin 4 &
tsh> jobs
[1] (10599) Running ./myspin 4 &
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$
```

```
终端
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$ make
./sdriver.pl -t trace14.txt -s ./tshref -a "-p"
#
# trace14.txt - Simple error handling
#
tsh> ./bogus
./bogus: Command not found
tsh> ./myspin 4 &
[1] (10653) ./myspin 4 &
tsh> fg
fg command requires PID or %jobid argument
tsh> bg
bg command requires PID or %jobid argument
tsh> fg a
fg: argument must be a PID or %jobid
tsh> bg a
bg: argument must be a PID or %jobid
tsh> fg 9999999
(9999999): No such process
tsh> bg 9999999
(9999999): No such process
tsh> fg %2
%2: No such job
tsh> fg %1
Job [1] (10653) stopped by signal 20
tsh> bg %2
%2: No such job
tsh> bg %1
[1] (10653) ./myspin 4 &
tsh> jobs
[1] (10653) Running ./myspin 4 &
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$
```

测试结论	相同

4.3.15 测试用例 trace15.txt 的输出截图（1 分）

tsh 测试结果	tshref 测试结果
----------	-------------

```

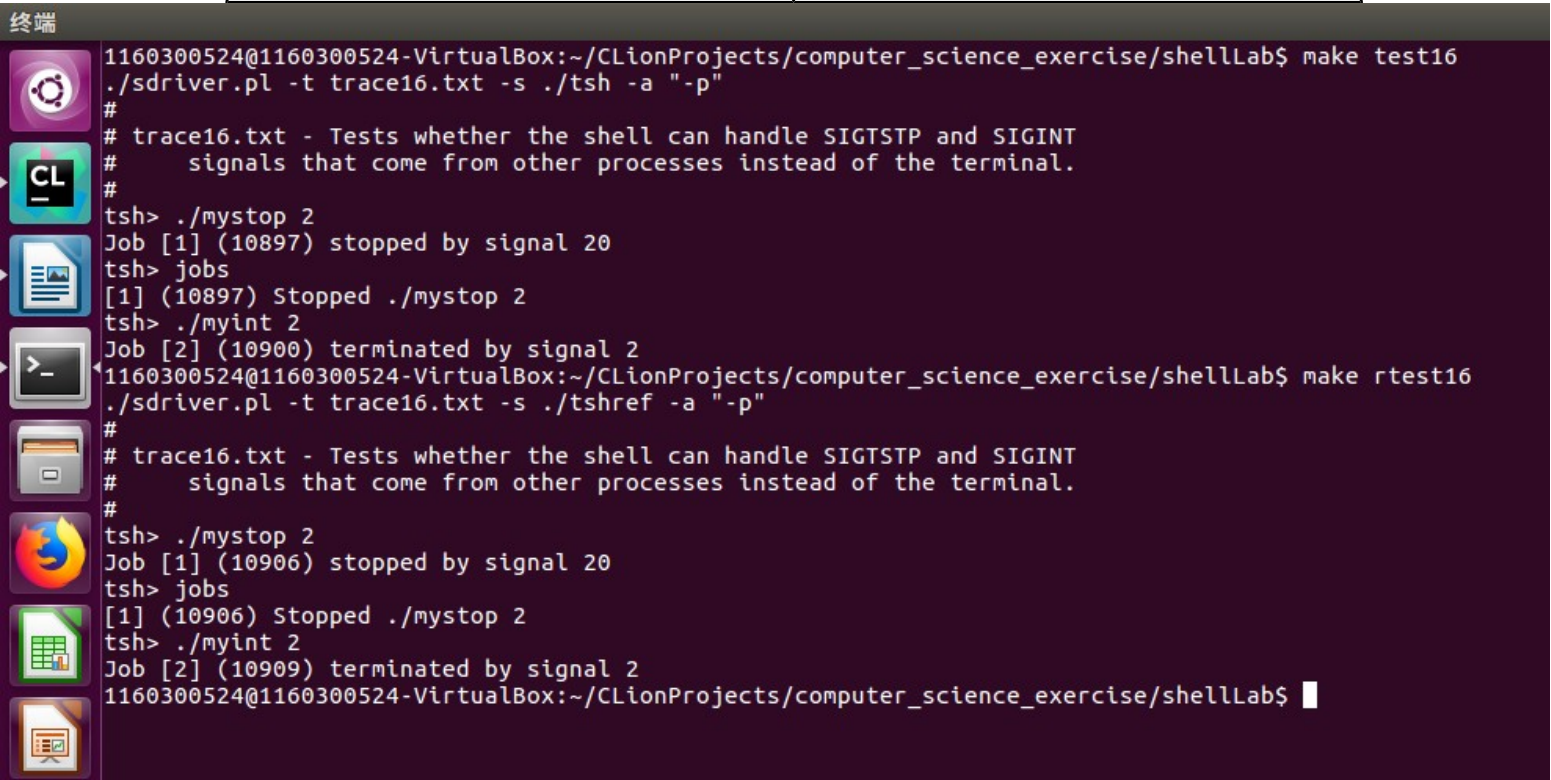
#
tsh> ./bogus
./bogus: Command not found
tsh> ./myspin 10
Job [1] (10829) terminated by signal 2
tsh> ./myspin 3 &
[1] (10831) ./myspin 3 &
tsh> ./myspin 4 &
[2] (10833) ./myspin 4 &
tsh> jobs
[1] (10831) Running ./myspin 3 &
[2] (10833) Running ./myspin 4 &
tsh> fg %1
Job [1] (10831) stopped by signal 20
tsh> jobs
[1] (10831) Stopped ./myspin 3 &
[2] (10833) Running ./myspin 4 &
tsh> bg %3
%3: No such job
tsh> bg %1
[1] (10831) ./myspin 3 &
tsh> jobs
[1] (10831) Running ./myspin 3 &
[2] (10833) Running ./myspin 4 &
tsh> fg %1
tsh> quit
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$ make rtest15
./sdriver.pl -t trace15.txt -s ./tshref -a "-p"
#
# trace15.txt - Putting it all together
#
tsh> ./bogus
./bogus: Command not found
tsh> ./myspin 10
Job [1] (10849) terminated by signal 2
tsh> ./myspin 3 &
[1] (10851) ./myspin 3 &
tsh> ./myspin 4 &
[2] (10853) ./myspin 4 &
tsh> jobs
[1] (10851) Running ./myspin 3 &
[2] (10853) Running ./myspin 4 &
tsh> fg %1
Job [1] (10851) stopped by signal 20
tsh> jobs
[1] (10851) Stopped ./myspin 3 &
[2] (10853) Running ./myspin 4 &
tsh> bg %3
%3: No such job
tsh> bg %1
[1] (10851) ./myspin 3 &
tsh> jobs
[1] (10851) Running ./myspin 3 &
[2] (10853) Running ./myspin 4 &
tsh> fg %1
tsh> quit
1160300524@1160300524-VirtualBox:~/CLionProjects/computer_science_exercise/shellLab$

```

测试结论	相同

4.3.16 测试用例 trace16.txt 的输出截图（1 分）

tsh 测试结果	tshref 测试结果
----------	-------------



测试结论	相同

4.4 自测试评分

根据节 4.3 的自测试结果，程序的测试评分为：16。

第 4 章 总结

4.1 请总结本次实验的收获

充分了解了 linux 系统的创建回收进程的机制，消息传送，处理机制，对于并发多进程编程有了充分的理解，自己认真编写了一个简易的 shell,感觉还是很有成就感的。

4.2 请给出对本次实验内容的建议

注：本章为酌情加分项。

参考文献

为完成本次实验你翻阅的书籍与网站等

- [1] 林来兴. 空间控制技术[M]. 北京: 中国宇航出版社, 1992: 25-42.
- [2] 辛希孟. 信息技术与信息服务国际研讨会论文集: A 集[C]. 北京: 中国科学出版社, 1999.
- [3] 赵耀东. 新时代的工业工程师[M/OL]. 台北: 天下文化出版社, 1998 []. <http://www.ie.nthu.edu.tw/info/ie.newie.htm> (Big5) .
- [4] 湛颖. 空间交会控制理论与方法研究[D]. 哈尔滨: 哈尔滨工业大学, 1992: 8-13.
- [5] KANAMORI H. Shaking Without Quaking[J]. Science , 1998 , 279 (5359) : 2063-2064.
- [6] CHRISTINE M. Plant Physiology: Plant Biology in the Genome Era[J/OL]. Science , 1998 , 281 : 331-332[]. <http://www.sciencemag.org/cgi/collection/anatmorp>.