

# 哈尔滨工业大学

# 实验报告

## 实验（七）

题    目 TinyShell  
微壳

专    业 计算机系

学    号 1180300811

班    级 1803008

学    生 孙骁

指 导 教 师 吴锐

实 验 地 点 G712

实 验 日 期 2019/12/07

计算机科学与技术学院

# 目 录

<b>第 1 章 实验基本信息</b>	<b>- 4 -</b>
1.1 实验目的	- 4 -
1. 理解现代计算机系统进程与并发的基本知识	- 4 -
2. 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	- 4 -
3. 掌握 SHELL 的基本原理和实现方法	- 4 -
4. 深入理解 LINUX 信号响应可能导致的并发冲突及解决方法	- 4 -
5. 培养 LINUX 下的软件系统开发与测试能力	- 4 -
1.2 实验环境与工具	- 4 -
1.2.1 硬件环境	- 4 -
1.2.2 软件环境	- 4 -
1.2.3 开发工具	- 4 -
1.3 实验预习	- 4 -
<b>第 2 章 实验预习</b>	<b>- 5 -</b>
2.1 进程的概念、创建和回收方法（5 分）	- 5 -
2.2 信号的机制、种类（5 分）	- 5 -
2.3 信号的发送方法、阻塞方法、处理程序的设置方法（5 分）	- 7 -
2.4 什么是 SHELL，功能和处理流程（5 分）	- 8 -
<b>第 3 章 TINY SHELL 的设计与实现</b>	<b>- 10 -</b>
3.1.1 VOID EVAL(CHAR *CMDLINE)函数（10 分）	- 10 -
3.1.2 INT BUILTIN_CMD(CHAR **ARGV)函数（5 分）	- 10 -
3.1.3 VOID DO_BGFG(CHAR **ARGV) 函数（5 分）	- 11 -
3.1.4 VOID WAITFG(PID_T PID) 函数（5 分）	- 11 -
3.1.5 VOID SIGCHLD_HANDLER(INT SIG) 函数（10 分）	- 12 -
<b>第 4 章 TINY SHELL 测试</b>	<b>- 40 -</b>
4.1 测试方法	- 40 -
4.2 测试结果评价	- 40 -
4.3 自测试结果	- 40 -
4.3.1 测试用例 trace01.txt 的输出截图（1 分）	- 40 -
4.3.2 测试用例 trace02.txt 的输出截图（1 分）	- 41 -
4.3.3 测试用例 trace03.txt 的输出截图（1 分）	- 41 -
4.3.4 测试用例 trace04.txt 的输出截图（1 分）	- 41 -
4.3.5 测试用例 trace05.txt 的输出截图（1 分）	- 42 -
4.3.6 测试用例 trace06.txt 的输出截图（1 分）	- 42 -
4.3.7 测试用例 trace07.txt 的输出截图（1 分）	- 43 -
4.3.8 测试用例 trace08.txt 的输出截图（1 分）	- 44 -

4.3.9 测试用例 <i>trace09.txt</i> 的输出截图 (1 分)	- 45 -
4.3.10 测试用例 <i>trace10.txt</i> 的输出截图 (1 分)	- 46 -
4.3.11 测试用例 <i>trace11.txt</i> 的输出截图 (1 分)	- 47 -
4.3.12 测试用例 <i>trace12.txt</i> 的输出截图 (1 分)	- 49 -
4.3.13 测试用例 <i>trace13.txt</i> 的输出截图 (1 分)	- 51 -
4.3.14 测试用例 <i>trace14.txt</i> 的输出截图 (1 分)	- 54 -
4.3.15 测试用例 <i>trace15.txt</i> 的输出截图 (1 分)	- 55 -
4.4 自测试评分	- 56 -
<b>第 4 章 总结</b>	<b>- 57 -</b>
4.1 请总结本次实验的收获	- 57 -
4.2 请给出对本次实验内容的建议	- 57 -
<b>参考文献</b>	<b>- 58 -</b>

## 第 1 章 实验基本信息

### 1.1 实验目的

1. 理解现代计算机系统进程与并发的基本知识
2. 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
3. 掌握 shell 的基本原理和实现方法
4. 深入理解 Linux 信号响应可能导致的并发冲突及解决方法
5. 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

i7-8550U X64 CPU; 1.80GHz; 16G RAM; 1T SSD

#### 1.2.2 软件环境

Windows10 64 位; Vmware 15.1.0; Ubuntu 18.04 LTS

#### 1.2.3 开发工具

Visual Studio 2019 ; CodeBlocks; gcc

### 1.3 实验预习

1. 上实验课前, 必须认真预习实验指导书 (PPT 或 PDF)
2. 了解实验的目的、实验环境与软硬件工具、实验操作步骤, 复习与实验有关的理论知识。
3. 了解进程、作业、信号的基本概念和原理
4. 了解 shell 的基本原理
5. 熟知进程创建、回收的方法和相关系统函数
6. 熟知信号机制和信号处理相关的系统函数

## 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）

#### 1. 概念：

进程是一个执行中程序的实例。系统中的每个程序都运行在某个进程的上下文中。

#### 2. 创建方法：

每次用户通过向 shell 输入一个可执行目标文件的名字，运行程序时，shell 就会创建一个新的进程，然后在这个新进程的上下文中运行这个可执行目标文件。

父进程通过调用 fork 函数创建一个新的运行的子进程

#### 3. 回收方法：

进程因三个原因终止：

- a) 收到一个信号，该信号的默认行为是终止进程
- b) 从主程序返回
- c) 调用 exit 函数

当一个进程由于某种原因终止时，内核并不是把它立即从系统中清除。相反，进程被保持在一种已终止的状态中，知道被它的父进程回收。当父进程回收已终止的子进程时，内核将子进程的退出状态传递给父进程，然后抛弃已终止的进程，从此开始，该进程就不存在了。

一个终止了但还未被回收的进程称为僵死进程。若父进程终止了，内核会安排 init 进程称为它的孤儿的养父，init 进程的 PID 为 1，是在系统启动时由内核创建的，不会终止，是所有进程的祖先。

### 2.2 信号的机制、种类（5 分）

#### 1. 信号的机制

信号机制是一种让信号在内核和进程之间传递信息的方法，全称是软中断信号，简称软中断。包括内核如何向一个进程发送信号、进程如何接收一个信号、进程怎样控制自己对信号的反应、内核在什么时机处理和怎

样处理进程收到的信号。

信号的处理方式：

Linux 的每一个信号都有一个缺省的动作，典型的缺省动作是终止进程，当一个信号到来的时候收到这个信号的进程会根据信号的具体情况提供一下三种不同的处理方式：

- 1) 对于需要处理的信号，指定处理函数，由该函数来处理。
- 2) 忽略某个信号，对该信号不做任何处理。
- 3) 对该信号的处理保留系统的默认值，这种缺省操作大多数使得进程终止，进程通过系统调用 `signal` 函数来指定进程对某个信号的处理行为。

## 2. 信号的种类

终端下运行命令 `kill -l` 得到系统定义的信号列表

```
demerzel@demerzel-virtual-machine:~$ kill -l
1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP
6) SIGABRT     7) SIGBUS     8) SIGFPE      9) SIGKILL     10) SIGUSR1
11) SIGSEGV    12) SIGUSR2    13) SIGPIPE    14) SIGALRM    15) SIGTERM
16) SIGSTKFLT  17) SIGCHLD   18) SIGCONT    19) SIGSTOP    20) SIGTSTP
21) SIGTTIN    22) SIGTTOU   23) SIGURG     24) SIGXCPU    25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF   28) SIGWINCH   29) SIGIO       30) SIGPWR
31) SIGSYS     34) SIGRTMIN  35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
```

1-31 号信号是普通信号，34-64 号信号是实时信号。无 32 和 33 号信号。

通过指令 `kill 信号序号 进程号` 可以向一个进程发送信号

序号	名称	默认行为	相应事件
1	SIGHUP	终止	终端线挂断
2	SIGINT	终止	来自键盘的中断
3	SIGQUIT	终止	来自键盘的退出
4	SIGILL	终止	非法指令
5	SIGTRAP	终止并转储内存	跟踪陷阱
6	SIGABRT	终止并转储内存	来自 <code>abort</code> 函数的终止信号
7	SIGBUS	终止	总线错误
8	SIGFPE	终止并转储内存	浮点异常
9	SIGKILL	终止	杀死程序
10	SIGUSR1	终止	用户定义的信号 1
11	SIGSEGV	终止并转储内存	无效的内存引用（段错误）
12	SIGUSR2	终止	用户定义的信号 2
13	SIGPIPE	终止	向一个没有读用户的管道做写操作

14	SIGALRM	终止	来自 alarm 函数的定时器信号
15	SIGTERM	终止	软件终止信号
16	SIGSTKFLT	终止	协处理器上的栈故障
17	SIGCHLD	忽略	一个子程序停止或者终止
18	SIGCONT	忽略	继续进程如果该进程停止
19	SIGSTOP	停止直到下一个 SIGCONT	不是来自终端的终止信号
20	SIGTSTP	停止直到下一个 SIGCONT	来自终端的终止信号
21	SIGTTIN	停止直到下一个 SIGCONT	后台进程从终端读
22	SIGTTOU	停止直到下一个 SIGCONT	后台进程向终端写
23	SIGURG	忽略	套接字上的紧急情况
24	SIGXCPU	终止	CPU 时间限制超出
25	SIGXFSZ	终止	文件大小限制超出
26	SIGVTALRM	终止	虚拟定时器期满
27	SIGPROF	终止	剖析定时器期满
28	SIGWINCH	忽略	窗口大小变化
29	SIGIO	终止	在某个描述符上可执行 I/O 操作
30	SIGPWR	终止	电源故障

## 2.3 信号的发送方法、阻塞方法、处理程序的设置方法（5 分）

内核通过更新目的进程上下文中的某个状态，发送（递送）一个信号给目的进程。

### 1. 发送方法：

#### a) 用 /bin/kill 程序发送信号

/bin/kill 程序可以向另外的进程或进程组发送任意的信号。

Examples: /bin/kill -9 24818 发送信号 9 (SIGKILL) 给进程 24818

/bin/kill -9 -24817 发送信号 SIGKILL 给进程组 24817 中的每个进程（负的 PID 会导致信号被发送到进程组 PID 中的每个进程）。

#### b) 从键盘发送信号输入 ctrl-c (ctrl-z) 会导致内核发送一个 SIGINT (SIGTSTP) 信号到前台进程组中的每个作业 SIGINT 默认情况是终止前台作业 SIGTSTP 默认情况是停止（挂起）前台作业。

#### c) 发送信号的函数主要有 kill(),raise(),alarm(),pause()

kill()和 raise()

kill()函数和熟知的 kill 系统命令一样，可以发送信号给信号和进程组（实际上 kill 系统命令只是 kill 函数的一个用户接口），需要注意的是他不仅可以终止进程(发送 SIGKILL 信号)，也可以向进程发送其他信号。与 kill

函数不同的是 `raise()` 函数允许进程向自身发送信号。

## 2. 阻塞信号

分为阻塞和解除阻塞信号：

### 1) 隐式阻塞机制：

内核默认阻塞与当前正在处理信号类型相同的待处理信号 如： 一个 `SIGINT` 信号处理程序不能被另一个 `SIGINT` 信号中断 （此时另一个 `SIGINT` 信号被阻塞）。

### 2) 显示阻塞和解除阻塞机制：

`sigprocmask` 函数及其辅助函数可以明确地阻塞/解除阻塞选定的信号辅助函数：

`sigemptyset` - 初始化 `set` 为空集合

`sigfillset` - 把每个信号都添加到 `set` 中

`sigaddset` - 把指定的信号 `signum` 添加到 `set`

`sigdelset` - 从 `set` 中删除指定的信号 `signum`

## 3. 设置信号处理程序

可以使用 `signal` 函数修改和信号 `signum` 相关联的默认行为: `handler_t`

`*signal(int signum, handler_t *handler)`

`handler` 的不同取值：

a) `SIG_IGN`: 忽略类型为 `signum` 的信号；

b) `SIG_DFL`: 类型为 `signum` 的信号行为恢复为默认行为；

c) 否则， `handler` 就是用户定义的函数的地址，这个函数称为信号处理程序；

只要进程接收到类型为 `signum` 的信号就会调用信号处理程序。

将处理程序的地址传递到 `signal` 函数从而改变默认行为，这叫作设置信号处理程序。调用信号处理程序称为捕获信号。

执行信号处理程序称为处理信号，当处理程序执行 `return` 时，控制会传递到控制流中被信号接收所中断的指令处。

## 2.4 什么是 shell，功能和处理流程（5 分）

### 1. shell 定义

shell 是系统的用户界面，提供了用户与内核进行交互操作的一种接口。可



以接收用户输入的命令并将其送入内核去执行。

## 2. shell 功能

shell 是一个命令解释器，它解释由用户输入的命令并且把它们送到内核。不仅如此，shell 有自己的编程语言用于对命令的编辑，它允许用户编写由 shell 命令组成的程序。

## 3. shell 处理流程

shell 首先检查命令是否是内部命令，若不是再检查是否是一个应用程序（可以是 Linux 本身的实用程序，如 `ls` 和 `rm`，也可以是购买的商业程序，或者是自由软件）。然后 shell 在可执行程序目录列表里寻找这些应用程序。如果键入的命令不是一个内部命令并且在列表里没有找到这个可执行文件，将会显示一条错误信息。如果能够成功找到命令，该内部命令或应用程序将被分解为系统调用并传给 Linux 内核。

## 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：解析命令，运行程序

参 数：从键盘输入的字符串的首地址

处理流程：

1. 定义各个变量。使用 `parseline()` 函数解析命令行，得到命令行参数。
2. 使用 `builtin_cmd()` 函数判断命令是否为内置命令，如果不是内置命令，则继续执行。
3. 设置阻塞集合。先初始化 `mask` 为空集合，再将 `SIGCHLD` , `SIGINT` , `SIGTSTP` 信号加入阻塞集合。
4. 阻塞 `SIGCHLD`，防止子进程在父进程之前结束，防止 `addjob()` 函数错误地把（不存在的）子进程添加到作业列表中。
5. 子进程中，先解除对 `SIG_CHLD` 的阻塞，再使用 `setpgid(0,0)` 创建一个虚拟的进程组，进程组 ID 是 15213，不和 `tsh` 进程在一个进程组。然后调用 `execve` 函数，执行相应的文件。
6. 将 `job` 添加到 `job list`，解除 `SIG_CHLD` 阻塞信号。判断进程是否为前台进程，如果是前台进程，调用 `waitfg()` 函数，等待前台进程，如果是后台进程，则打印出进程信息。

要点分析：

在添加新进程，即 `fork` 之前要阻塞信号，防止 `addjob()` 函数错误地把（不存在的）子进程添加到作业列表中。`fork` 之后，在子进程开始时解除阻塞；在父进程添加完 `job` 信息后解除阻塞。

#### 3. 1.2 int builtin\_cmd(char \*\*argv) 函数 (5 分)

函数功能：判断用户输入的是否为内置命令，若是，则执行。

参 数：终端输入字符串整理后的字符数组

处理流程：

根据 `argv[0]` 判断用户输入的命令类型，若为内置命令，执行相应的命令，返回 1。`quit` 相当于退出 `shell`，直接调用 `exit(0)`；`jobs` 为打印当前 `job` 信息，调用 `listjobs`，返回 1；`fg` 与 `bg` 都是控制运行前台和后台程序，调用 `do_bgfg` 函数；不是内置命

令就返回 0。

要点分析：

`jobs` 是全局变量，为了防止其被修改，需要阻塞全部信号，过程大致为（后面函数阻塞全部信号的做法与此基本一致）：

```
sigset_t mask, prev_mask;

sigfillset(&mask);

sigprocmask(SIG_BLOCK, &mask, &prev_mask);

listjobs(jobs);

sigprocmask(SIG_SETMASK, &prev_mask, NULL);
```

### 3. 1.3 void do\_bgfg(char \*\*argv) 函数（5 分）

函数功能：执行 `bg` 或者 `fg` 命令

参 数：终端输入字符串整理后的字符数组

处理流程：

1. 判断输入的第二个参数是否为空，即 `argv[1]` 是否为空，若为空，则输出错误。
2. 根据输入的 `PID` 或者 `%JID` 判断 `jobs` 里面是否有相应信息，没有则输出相应提示。
3. 如果有，则根据 `bg` 或者 `fg` 向进程组发送 `SIGCONT`，如果是 `bg`，则输出相应 `job` 信息，如果是 `fg` 则等待 `job` 终止。

要点分析：

函数先判断 `fg` 后面是 `%+数字` 还是只有数字的形式，从而根据进程号 `PID` 或是工作组号 `%JID` 来获取 `job`；然后在根据前台和后台进程的不同，分别执行相应的操作。

### 3. 1.4 void waitfg(pid\_t pid) 函数（5 分）

函数功能：等待 `PID` 对应的进程不再是前台程序

参 数：前台程序的 `PID`

处理流程：

1. 调用 `fgpid` 获取前台程序 `PID` 并和参数的 `PID` 比较
2. 若相等，循环，执行 `sleep(0)`;
3. 若不等，返回。

要点分析：

`sleep` 函数参数设置为 0，可将 CPU 占用转移到其他需要的进程。

### 3. 1.5 void sigchld\_handler(int sig) 函数 (10 分)

函数功能：回收因父进程结束而产生的所有僵尸子进程。

参 数：

形参：信号 sig；实参 status, WNOHANG|WUNTRACED, SIG\_BLOCK, mask, prev\_mask, SIG\_SETMASK

处理流程：

1. 将所有信号添加到 mask 阻塞集中, 设置 olderrno = errno, 保存旧的 errno。
2. while 循环尽可能收集子进程。若有停止或者终止的, 返回其 PID, 没有则返回 0。
3. 循环中阻塞信号, 调用 getjobpid() 函数, 通过 PID 找到 job。
4. 通过 waitpid 函数设置 status 中子进程的状态信息, 判断子进程的退出状态。如果子进程当前是停止的, 那么 WIFSTOPPED(status) 就返回 1, 此时只需要将 PID 找到的 job 的状态改为 ST, 并且输出 job 的 jid, pid 以及导致子进程停止的信号编号。如果子进程是因为一个未被捕获的信号终止的, 那么 WIFSIGNALED(status) 就返回 1, 输出 job 的 jid, pid 以及导致子进程终止的信息编号, 因为此时进程是中止的进程, 所以还需要 deletejob() 将发出 SIGCHLD 信号的进程回收。
5. 清空缓冲区, 恢复旧的 errno。

要点分析：

1. while 循环避免信号阻塞, 使用 waitpid() 函数, 以尽可能多的回收僵尸进程。
2. 调用 deletejob() 函数时, 因为 jobs 是全局变量, 因此需要阻塞信号。
3. 通过 waitpid 设置 status 的返回子进程的状态信息, 判断子进程的退出状态。
4. WIFSIGNALED 判断子进程是否为一个未被捕获的信号中止的, WIFSTOPPED 判断子进程当前是否为停止的。

### 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

```
/*
 * tsh - A tiny shell program with job control
 *
 * <Put your name and login ID here>
 */
#include <stdio.h>
```

```
#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <ctype.h>

#include <signal.h>

#include <sys/types.h>

#include <sys/wait.h>

#include <errno.h>


/* Misc manifest constants */

#define MAXLINE    1024    /* max line size */
#define MAXARGS    128    /* max args on a command line */
#define MAXJOBS    16     /* max jobs at any point in time */
#define MAXJID     1<<16  /* max job ID */


/* Job states */

#define UNDEF 0 /* undefined */

#define FG 1    /* running in foreground */
#define BG 2    /* running in background */
#define ST 3    /* stopped */


/*

* Jobs states: FG (foreground), BG (background), ST (stopped)

* Job state transitions and enabling actions:

*     FG -> ST   : ctrl-z

*     ST -> FG   : fg command
```

```

*      ST -> BG   : bg command

*      BG -> FG   : fg command

* At most 1 job can be in the FG state.

*/

/* Global variables */

extern char **environ;      /* defined in libc */

char prompt[] = "tsh> ";   /* command line prompt (DO NOT CHANGE) */

int verbose = 0;           /* if true, print additional output */

int nextjid = 1;           /* next job ID to allocate */

char sbuf[MAXLINE];        /* for composing sprintf messages */

struct job_t {              /* The job struct */
    pid_t pid;              /* job PID */
    int jid;                /* job ID [1, 2, ...] */
    int state;              /* UNDEF, BG, FG, or ST */
    char cmdline[MAXLINE];  /* command line */
};

struct job_t jobs[MAXJOBS]; /* The job list */

/* End global variables */

/* Function prototypes */

/* Here are the functions that you will implement */

void eval(char *cmdline);

```

```
int builtin_cmd(char **argv);

void do_bgfg(char **argv);

void waitfg(pid_t pid);


void sigchld_handler(int sig);
void sigtstp_handler(int sig);
void sigint_handler(int sig);


/* Here are helper routines that we've provided for you */

int parseline(const char *cmdline, char **argv);
void sigquit_handler(int sig);


void clearjob(struct job_t *job);
void initjobs(struct job_t *jobs);
int maxjid(struct job_t *jobs);
int addjob(struct job_t *jobs, pid_t pid, int state, char *cmdline);
int deletejob(struct job_t *jobs, pid_t pid);
pid_t fgpid(struct job_t *jobs);
struct job_t *getjobpid(struct job_t *jobs, pid_t pid);
struct job_t *getjobjid(struct job_t *jobs, int jid);
int pid2jid(pid_t pid);
void listjobs(struct job_t *jobs);


void usage(void);
void unix_error(char *msg);
void app_error(char *msg);
```

```
typedef void handler_t(int);

handler_t *Signal(int signum, handler_t *handler);

/*
 * main - The shell's main routine
 */
int main(int argc, char **argv)
{
    char c;

    char cmdline[MAXLINE];

    int emit_prompt = 1; /* emit prompt (default) */

    /* Redirect stderr to stdout (so that driver will get all output
     * on the pipe connected to stdout) */
    dup2(1, 2);

    /* Parse the command line */
    while ((c = getopt(argc, argv, "hvp")) != EOF) {
        switch (c) {
            case 'h':          /* print help message */
                usage();
                break;

            case 'v':          /* emit additional diagnostic info */
                verbose = 1;
                break;

            case 'p':          /* don't print a prompt */
```



```
        emit_prompt = 0; /* handy for automatic testing */

        break;

default:

        usage();

}

}

/* Install the signal handlers */

/* These are the ones you will need to implement */

Signal(SIGINT,  sigint_handler); /* ctrl-c */

Signal(SIGTSTP, sigtstp_handler); /* ctrl-z */

Signal(SIGCHLD, sigchld_handler); /* Terminated or stopped child */

/* This one provides a clean way to kill the shell */

Signal(SIGQUIT, sigquit_handler);

/* Initialize the job list */

initjobs(jobs);

/* Execute the shell's read/eval loop */

while (1) {

/* Read command line */

if (emit_prompt) {

        printf("%s", prompt);
```

```
        fflush(stdout);
    }
    if ((fgets(cmdline, MAXLINE, stdin) == NULL) && ferror(stdin))
        app_error("fgets error");
    if (feof(stdin)) { /* End of file (ctrl-d) */
        fflush(stdout);
        exit(0);
    }

    /* Evaluate the command line */
    eval(cmdline);
    fflush(stdout);
    fflush(stdout);
}

exit(0); /* control never reaches here */
}

/*
 * eval - Evaluate the command line that the user has just typed in
 *
 * If the user has requested a built-in command (quit, jobs, bg or fg)
 * then execute it immediately. Otherwise, fork a child process and
 * run the job in the context of the child. If the job is running in
 * the foreground, wait for it to terminate and then return.  Note:
 * each child process must have a unique process group ID so that our
```

```

* background children don't receive SIGINT (SIGTSTP) from the kernel
* when we type ctrl-c (ctrl-z) at the keyboard.
*/

void eval(char *cmdline)
{
    /* $begin handout */

    char *argv[MAXARGS]; /* argv for execve() */

    int bg;                /* should the job run in bg or fg? */
    pid_t pid;             /* process id */
    sigset_t mask;         /* signal mask */

    /* Parse command line */

    bg = parseline(cmdline, argv);
    if (argv[0] == NULL)
        return; /* ignore empty lines */

    if (!builtin_cmd(argv)) {

        /*

        * This is a little tricky. Block SIGCHLD, SIGINT, and SIGTSTP
        * signals until we can add the job to the job list. This
        * eliminates some nasty races between adding a job to the job
        * list and the arrival of SIGCHLD, SIGINT, and SIGTSTP signals.
        */

        if (sigemptyset(&mask) < 0)

```

```
        unix_error("sigemptyset error");

    if (sigaddset(&mask, SIGCHLD))
        unix_error("sigaddset error");

    if (sigaddset(&mask, SIGINT))
        unix_error("sigaddset error");

    if (sigaddset(&mask, SIGTSTP))
        unix_error("sigaddset error");

    if (sigprocmask(SIG_BLOCK, &mask, NULL) < 0)
        unix_error("sigprocmask error");


    /* Create a child process */
    if ((pid = fork()) < 0)
        unix_error("fork error");


    /*
     * Child process
     */

    if (pid == 0) {
        /* Child unblocks signals */
        sigprocmask(SIG_UNBLOCK, &mask, NULL);


        /* Each new job must get a new process group ID
           so that the kernel doesn't send ctrl-c and ctrl-z
           signals to all of the shell's jobs */
        if (setpgid(0, 0) < 0)
```

```

        unix_error("setpgid error");

        /* Now load and run the program in the new job */
        if (execve(argv[0], argv, environ) < 0) {
            printf("%s: Command not found\n", argv[0]);
            exit(0);
        }
    }

    /*
     * Parent process
     */

    /* Parent adds the job, and then unblocks signals so that
       the signals handlers can run again */
    addjob(jobs, pid, (bg == 1 ? BG : FG), cmdline);
    sigprocmask(SIG_UNBLOCK, &mask, NULL);

    if (!bg)
        waitfg(pid);
    else
        printf("[%d] (%d) %s", pid2jid(pid), pid, cmdline);
    }

    /* $end handout */

    return;
}

```

```
/*
 * parseline - Parse the command line and build the argv array.
 *
 * Characters enclosed in single quotes are treated as a single
 * argument.  Return true if the user has requested a BG job, false if
 * the user has requested a FG job.
 */
int parseline(const char *cmdline, char **argv)
{
    static char array[MAXLINE]; /* holds local copy of command line */
    char *buf = array;          /* ptr that traverses command line */
    char *delim;                 /* points to first space delimiter */
    int argc;                    /* number of args */
    int bg;                      /* background job? */

    strcpy(buf, cmdline);
    buf[strlen(buf)-1] = ' '; /* replace trailing '\n' with space */
    while (*buf && (*buf == ' ')) /* ignore leading spaces */
        buf++;

    /* Build the argv list */
    argc = 0;
    if (*buf == "\\") {
        buf++;
        delim = strchr(buf, "\\");
    }
```

```
    }

    else {
        delim = strchr(buf, ' ');
    }

    while (delim) {
        argv[argc++] = buf;
        *delim = '\0';
        buf = delim + 1;
        while (*buf && (*buf == ' ')) /* ignore spaces */
            buf++;

        if (*buf == "\\") {
            buf++;
            delim = strchr(buf, "\\");
        }
        else {
            delim = strchr(buf, ' ');
        }
    }

    argv[argc] = NULL;

    if (argc == 0) /* ignore blank line */
        return 1;

    /* should the job run in the background? */
```

```
    if ((bg = (*argv[argc-1] == '&')) != 0) {
        argv[--argc] = NULL;
    }
    return bg;
}

/*
 * builtin_cmd - If the user has typed a built-in command then execute
 *               it immediately.
 */
int builtin_cmd(char **argv)
{
    sigset_t mask, prev_mask;
    sigfillset(&mask);
    if (strcmp(argv[0], "quit") == 0)
        exit(0);
    if (strcmp(argv[0], "jobs") == 0) {
        sigprocmask(SIG_BLOCK, &mask, &prev_mask);
        listjobs(jobs);
        sigprocmask(SIG_SETMASK, &prev_mask, NULL);
        return 1;
    }
    if (strcmp(argv[0], "bg") == 0 || strcmp(argv[0], "fg") == 0) {
        do_bgfg(argv);
        return 1;
    }
}
```



```
    if (strcmp(argv[0], "&") == 0)
        return 1;
    return 0;        /* not a builtin command */
}

/*
 * do_bgfg - Execute the builtin bg and fg commands
 */
void do_bgfg(char **argv)
{
    /* $begin handout */
    struct job_t *jobp=NULL;

    /* Ignore command if no argument */
    if (argv[1] == NULL) {
        printf("%s command requires PID or %%jobid argument\n", argv[0]);
        return;
    }

    /* Parse the required PID or %JID arg */
    if (isdigit(argv[1][0])) {
        pid_t pid = atoi(argv[1]);
        if (!(jobp = getjobpid(jobs, pid))) {
            printf("(%d): No such process\n", pid);
            return;
        }
    }
```

```
    }

    else if (argv[1][0] == '%') {
int jid = atoi(&argv[1][1]);
if (!(jobp = getjobjid(jobs, jid))) {
    printf("%s: No such job\n", argv[1]);
    return;
}
}

else {
printf("%s: argument must be a PID or %%jobid\n", argv[0]);
return;
}

/* bg command */
if (!strcmp(argv[0], "bg")) {
if (kill(-(jobp->pid), SIGCONT) < 0)
    unix_error("kill (bg) error");
jobp->state = BG;
printf("[%d] (%d) %s", jobp->jid, jobp->pid, jobp->cmdline);
}

/* fg command */
else if (!strcmp(argv[0], "fg")) {
if (kill(-(jobp->pid), SIGCONT) < 0)
    unix_error("kill (fg) error");
jobp->state = FG;
```

```

    waitfg(jobp->pid);
    }
    else {
        printf("do_bgfg: Internal error\n");
        exit(0);
    }
    /* $end handout */
    return;
}

/*
 * waitfg - Block until process pid is no longer the foreground process
 */
void waitfg(pid_t pid)
{
    while (fgpid(jobs) == pid)
        sleep(0);
    return;
}

/*****
 * Signal handlers
 *****/

/*
 * sigchld_handler - The kernel sends a SIGCHLD to the shell whenever

```

```
*      a child job terminates (becomes a zombie), or stops because it
*
*      received a SIGSTOP or SIGTSTP signal. The handler reaps all
*
*      available zombie children, but doesn't wait for any other
*
*      currently running children to terminate.
*/

void sigchld_handler(int sig)
{
    struct job_t* job_new;
    sigset_t mask, prev_mask;
    int olderrno = errno, status;
    pid_t pid;
    sigfillset(&mask);
    while ((pid = waitpid(-1, &status, WNOHANG | WUNTRACED)) > 0) {
        sigprocmask(SIG_BLOCK, &mask, &prev_mask);
        job_new = getjobpid(jobs, pid);
        if (WIFSTOPPED(status)) {
            job_new->state = ST;
            printf("Job [%d] (%d) terminated by signal %d\n", job_new->jid,
job_new->pid, WSTOPSIG(status));
        }
        else {
            if (WIFSIGNALED(status))
                printf("Job [%d] (%d) terminated by signal %d\n",
job_new->jid, job_new->pid, WTERMSIG(status));
            deletejob(jobs, pid);
        }
        fflush(stdout);
    }
}
```

```
        sigprocmask(SIG_SETMASK, &prev_mask, NULL);
    }

    errno = olderrno;
    return;
}

/*
 * sigint_handler - The kernel sends a SIGINT to the shell whenever the
 *      user types ctrl-c at the keyboard.  Catch it and send it along
 *      to the foreground job.
 */
void sigint_handler(int sig)
{
    int olderrno = errno;

    sigset_t mask, prev_mask;

    pid_t pid;

    sigfillset(&mask);

    sigprocmask(SIG_BLOCK, &mask, &prev_mask);

    pid = fgpid(jobs);

    sigprocmask(SIG_SETMASK, &prev_mask, NULL);

    if (pid != 0)
        kill(-pid, SIGINT);

    errno = olderrno;

    return;
}
```

```
/*
 * sigtstp_handler - The kernel sends a SIGTSTP to the shell whenever
 *                   the user types ctrl-z at the keyboard. Catch it and suspend the
 *                   foreground job by sending it a SIGTSTP.
 */

void sigtstp_handler(int sig)
{
    int olderrno = errno;
    sigset_t mask, prev_mask;
    pid_t pid;
    sigfillset(&mask);
    sigprocmask(SIG_BLOCK, &mask, &prev_mask);
    pid = fgpid(jobs);
    sigprocmask(SIG_SETMASK, &prev_mask, NULL);
    if (pid != 0)
        kill(-pid, SIGTSTP);
    errno = olderrno;
    return;
    return;
}

/*****

* End signal handlers

*****/

/*****
```

\* Helper routines that manipulate the job list

\*\*\*\*\*/

/\* clearjob - Clear the entries in a job struct \*/

void clearjob(struct job\_t \*job) {

    job->pid = 0;

    job->jid = 0;

    job->state = UNDEF;

    job->cmdline[0] = '\0';

}

/\* initjobs - Initialize the job list \*/

void initjobs(struct job\_t \*jobs) {

    int i;

    for (i = 0; i < MAXJOBS; i++)

        clearjob(&jobs[i]);

}

/\* maxjid - Returns largest allocated job ID \*/

int maxjid(struct job\_t \*jobs)

{

    int i, max=0;

    for (i = 0; i < MAXJOBS; i++)

        if (jobs[i].jid > max)

```
        max = jobs[i].jid;

    return max;

}

/* addjob - Add a job to the job list */

int addjob(struct job_t *jobs, pid_t pid, int state, char *cmdline)
{
    int i;

    if (pid < 1)
        return 0;

    for (i = 0; i < MAXJOBS; i++) {
        if (jobs[i].pid == 0) {
            jobs[i].pid = pid;
            jobs[i].state = state;
            jobs[i].jid = nextjid++;
            if (nextjid > MAXJOBS)
                nextjid = 1;
            strcpy(jobs[i].cmdline, cmdline);
            if(verbose){
                printf("Added job [%d] %d %s\n", jobs[i].jid, jobs[i].pid,
jobs[i].cmdline);
            }
            return 1;
        }
    }
}
```



```
    printf("Tried to create too many jobs\n");
    return 0;
}

/* deletejob - Delete a job whose PID=pid from the job list */
int deletejob(struct job_t *jobs, pid_t pid)
{
    int i;

    if (pid < 1)
        return 0;

    for (i = 0; i < MAXJOBS; i++) {
        if (jobs[i].pid == pid) {
            clearjob(&jobs[i]);
            nextjid = maxjid(jobs)+1;
            return 1;
        }
    }
    return 0;
}

/* fgpid - Return PID of current foreground job, 0 if no such job */
pid_t fgpid(struct job_t *jobs) {
    int i;
```

```
        for (i = 0; i < MAXJOBS; i++)
            if (jobs[i].state == FG)
                return jobs[i].pid;
        return 0;
    }

/* getjobpid - Find a job (by PID) on the job list */
struct job_t *getjobpid(struct job_t *jobs, pid_t pid) {
    int i;

    if (pid < 1)
        return NULL;
    for (i = 0; i < MAXJOBS; i++)
        if (jobs[i].pid == pid)
            return &jobs[i];
    return NULL;
}

/* getjobjid - Find a job (by JID) on the job list */
struct job_t *getjobjid(struct job_t *jobs, int jid)
{
    int i;

    if (jid < 1)
        return NULL;
    for (i = 0; i < MAXJOBS; i++)
```

```
    if (jobs[i].jid == jid)
        return &jobs[i];

    return NULL;
}

/* pid2jid - Map process ID to job ID */
int pid2jid(pid_t pid)
{
    int i;

    if (pid < 1)
        return 0;

    for (i = 0; i < MAXJOBS; i++)
        if (jobs[i].pid == pid) {
            return jobs[i].jid;
        }

    return 0;
}

/* listjobs - Print the job list */
void listjobs(struct job_t *jobs)
{
    int i;

    for (i = 0; i < MAXJOBS; i++) {
        if (jobs[i].pid != 0) {
```

```

        printf("[%d] (%d) ", jobs[i].jid, jobs[i].pid);
        switch (jobs[i].state) {
        case BG:
            printf("Running ");
            break;
        case FG:
            printf("Foreground ");
            break;
        case ST:
            printf("Stopped ");
            break;
        default:
            printf("listjobs: Internal error: job[%d].state=%d ",
                i, jobs[i].state);
        }
        printf("%s", jobs[i].cmdline);
    }
}

/*****

* end job list helper routines

*****/

/*****

* Other helper routines

```

```

*****/

/*
 * usage - print a help message
 */
void usage(void)
{
    printf("Usage: shell [-hvp]\n");
    printf("    -h    print this message\n");
    printf("    -v    print additional diagnostic information\n");
    printf("    -p    do not emit a command prompt\n");
    exit(1);
}

/*
 * unix_error - unix-style error routine
 */
void unix_error(char *msg)
{
    fprintf(stdout, "%s: %s\n", msg, strerror(errno));
    exit(1);
}

/*
 * app_error - application-style error routine
 */
```

```
void app_error(char *msg)
{
    fprintf(stdout, "%s\n", msg);
    exit(1);
}

/*
 * Signal - wrapper for the sigaction function
 */
handler_t *Signal(int signum, handler_t *handler)
{
    struct sigaction action, old_action;

    action.sa_handler = handler;

    sigemptyset(&action.sa_mask); /* block sigs of type being handled */
    action.sa_flags = SA_RESTART; /* restart syscalls if possible */

    if (sigaction(signum, &action, &old_action) < 0)
        unix_error("Signal error");

    return (old_action.sa_handler);
}

/*
 * sigquit_handler - The driver program can gracefully terminate the
 *     child shell by sending it a SIGQUIT signal.
 */
```

```
void sigquit_handler(int sig)
{
    printf("Terminating after receipt of SIGQUIT signal\n");
    exit(1);
}
```

## 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)

```
demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make test01
./sdriver.pl -t trace01.txt -s ./tsh -a "-p"
#
# trace01.txt - Properly terminate on EOF.
#
demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make rtest01
./sdriver.pl -t trace01.txt -s ./tshref -a "-p"
#
# trace01.txt - Properly terminate on EOF.
#
```

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）

```
demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make test02
./sdriver.pl -t trace02.txt -s ./tsh -a "-p"
#
# trace02.txt - Process builtin quit command.
#
demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make rtest02
./sdriver.pl -t trace02.txt -s ./tshref -a "-p"
#
# trace02.txt - Process builtin quit command.
#
```

tsh 测试结果		tshref 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）

```
demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make test03
./sdriver.pl -t trace03.txt -s ./tsh -a "-p"
#
# trace03.txt - Run a foreground job.
#
tsh> quit
demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make rtest03
./sdriver.pl -t trace03.txt -s ./tshref -a "-p"
#
# trace03.txt - Run a foreground job.
#
tsh> quit
```

tsh 测试结果		tshref 测试结果	
测试结论	相同		

## 4.3.4 测试用例 trace04.txt 的输出截图（1 分）

```
demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make test04
./sdriver.pl -t trace04.txt -s ./tsh -a "-p"
#
# trace04.txt - Run a background job.
#
tsh> ./myspin 1 &
[1] (33271) ./myspin 1 &
demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make rtest04
./sdriver.pl -t trace04.txt -s ./tshref -a "-p"
#
# trace04.txt - Run a background job.
#
tsh> ./myspin 1 &
[1] (33277) ./myspin 1 &
```

tsh 测试结果		tshref 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）

```

demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make test05
./sdriver.pl -t trace05.txt -s ./tsh -a "-p"
#
# trace05.txt - Process jobs builtin command.
#
tsh> ./myspin 2 &
[1] (33283) ./myspin 2 &
tsh> ./myspin 3 &
[2] (33285) ./myspin 3 &
tsh> jobs
[1] (33283) Running ./myspin 2 &
[2] (33285) Running ./myspin 3 &
demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make rtest05
./sdriver.pl -t trace05.txt -s ./tshref -a "-p"
#
# trace05.txt - Process jobs builtin command.
#
tsh> ./myspin 2 &
[1] (33292) ./myspin 2 &
tsh> ./myspin 3 &
[2] (33294) ./myspin 3 &
tsh> jobs
[1] (33292) Running ./myspin 2 &
[2] (33294) Running ./myspin 3 &

```

tsh 测试结果		tshref 测试结果	
测试结论	相同		

#### 4.3.6 测试用例 trace06.txt 的输出截图（1 分）

```

demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make test06
./sdriver.pl -t trace06.txt -s ./tsh -a "-p"
#
# trace06.txt - Forward SIGINT to foreground job.
#
tsh> ./myspin 4
Job [1] (33301) terminated by signal 2
demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make rtest06
./sdriver.pl -t trace06.txt -s ./tshref -a "-p"
#
# trace06.txt - Forward SIGINT to foreground job.
#
tsh> ./myspin 4
Job [1] (33307) terminated by signal 2

```

tsh 测试结果		tshref 测试结果	
测试结论	相同		

#### 4.3.7 测试用例 trace07.txt 的输出截图（1 分）

```

demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make test07
./sdriver.pl -t trace07.txt -s ./tsh -a "-p"
#
# trace07.txt - Forward SIGINT only to foreground job.
#
tsh> ./myspin 4 &
[1] (33340) ./myspin 4 &
tsh> ./myspin 5
Job [2] (33342) terminated by signal 2
tsh> jobs
[1] (33340) Running ./myspin 4 &
demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make rtest07
./sdriver.pl -t trace07.txt -s ./tshref -a "-p"
#
# trace07.txt - Forward SIGINT only to foreground job.
#
tsh> ./myspin 4 &
[1] (33349) ./myspin 4 &
tsh> ./myspin 5
Job [2] (33351) terminated by signal 2
tsh> jobs
[1] (33349) Running ./myspin 4 &

```

tsh 测试结果		tshref 测试结果	
测试结论	相同		

## 4.3.8 测试用例 trace08.txt 的输出截图（1 分）

```
demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make test08
./sdriver.pl -t trace08.txt -s ./tsh -a "-p"
#
# trace08.txt - Forward SIGTSTP only to foreground job.
#
tsh> ./myspin 4 &
[1] (33358) ./myspin 4 &
tsh> ./myspin 5
Job [2] (33360) terminated by signal 20
tsh> jobs
[1] (33358) Running ./myspin 4 &
[2] (33360) Stopped ./myspin 5
demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make rtest08
./sdriver.pl -t trace08.txt -s ./tshref -a "-p"
#
# trace08.txt - Forward SIGTSTP only to foreground job.
#
tsh> ./myspin 4 &
[1] (33367) ./myspin 4 &
tsh> ./myspin 5
Job [2] (33369) stopped by signal 20
tsh> jobs
[1] (33367) Running ./myspin 4 &
[2] (33369) Stopped ./myspin 5
```

tsh 测试结果		tshref 测试结果	
测试结论	相同		

## 4.3.9 测试用例 trace09.txt 的输出截图（1 分）

```

demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make test09
./sdriver.pl -t trace09.txt -s ./tsh -a "-p"
#
# trace09.txt - Process bg builtin command
#
tsh> ./myspin 4 &
[1] (33376) ./myspin 4 &
tsh> ./myspin 5
Job [2] (33378) terminated by signal 20
tsh> jobs
[1] (33376) Running ./myspin 4 &
[2] (33378) Stopped ./myspin 5
tsh> bg %2
[2] (33378) ./myspin 5
tsh> jobs
[1] (33376) Running ./myspin 4 &
[2] (33378) Running ./myspin 5
demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make rtest09
./sdriver.pl -t trace09.txt -s ./tshref -a "-p"
#
# trace09.txt - Process bg builtin command
#
tsh> ./myspin 4 &
[1] (33387) ./myspin 4 &
tsh> ./myspin 5
Job [2] (33389) stopped by signal 20
tsh> jobs
[1] (33387) Running ./myspin 4 &
[2] (33389) Stopped ./myspin 5
tsh> bg %2
[2] (33389) ./myspin 5
tsh> jobs
[1] (33387) Running ./myspin 4 &
[2] (33389) Running ./myspin 5

```

tsh 测试结果		tshref 测试结果	
测试结论	相同/不同，原因分析如下：		

## 4.3.10 测试用例 trace10.txt 的输出截图（1 分）

```
demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make test10
./sdriver.pl -t trace10.txt -s ./tsh -a "-p"
#
# trace10.txt - Process fg builtin command.
#
tsh> ./myspin 4 &
[1] (33398) ./myspin 4 &
tsh> fg %1
Job [1] (33398) terminated by signal 20
tsh> jobs
[1] (33398) Stopped ./myspin 4 &
tsh> fg %1
tsh> jobs
demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make rtest10
./sdriver.pl -t trace10.txt -s ./tshref -a "-p"
#
# trace10.txt - Process fg builtin command.
#
tsh> ./myspin 4 &
[1] (33408) ./myspin 4 &
tsh> fg %1
Job [1] (33408) stopped by signal 20
tsh> jobs
[1] (33408) Stopped ./myspin 4 &
tsh> fg %1
tsh> jobs
```

tsh 测试结果		tshref 测试结果	
测试结论	相同		

## 4.3.11 测试用例 trace11.txt 的输出截图（1 分）

```

demerzel@demerzel-virtual-machine:~/cocode/lab7/shlab-handout-hit$ 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] (33418) terminated by signal 2
tsh> /bin/ps a
  PID TTY          STAT TIME  COMMAND
 1200 tty1      Ssl+  0:00 /usr/lib/gdm3/gdm-wayland-session gnome-session --autostart /usr/share/gdm/greeter/autostart
 1239 tty1      Sl+   0:00 /usr/lib/gnome-session/gnome-session-binary --autostart /usr/share/gdm/greeter/autostart
 1401 tty1      Sl+   0:27 /usr/bin/gnome-shell
 1424 tty1      Sl+   9:40 /usr/bin/Xwayland :1024 -rootless -terminate -accessx -core -listen 4 -listen 5 -displayfd 6
 1465 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-xsettings
 1468 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-a11y-settings
 1472 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-clipboard
 1473 tty1      Sl+   0:21 /usr/lib/gnome-settings-daemon/gsd-color
 1474 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-datetime
 1476 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-housekeeping
 1484 tty1      Sl+   0:02 /usr/lib/gnome-settings-daemon/gsd-keyboard
 1487 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-media-keys
 1488 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-mouse
 1489 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-power
 1492 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-print-notifications
 1493 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-rfkill
 1494 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-screensaver-proxy
 1499 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-sharing
 1503 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-smartcard
 1504 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-sound
 1510 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-wacom
 1592 tty2      Ssl+  0:00 /usr/lib/gdm3/gdm-x-session --run-script env GNOME_SHELL_SESSION_MODE=ubuntu gnome-session --session=ubuntu
 1594 tty2      Sl+   6:45 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/1000/gdm/Xauthority -background none -noreset -keeptty -verbose 3
 1603 tty2      Sl+   0:01 /usr/lib/gnome-session/gnome-session-binary --session=ubuntu
 1739 tty2      Sl+   12:49 /usr/bin/gnome-shell
 1820 tty2      Sl+   0:01 /usr/lib/gnome-settings-daemon/gsd-power
 1821 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-print-notifications
 1823 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-rfkill
 1824 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-screensaver-proxy
 1829 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-sharing
 1841 tty2      Sl+   0:01 /usr/lib/gnome-settings-daemon/gsd-xsettings
 1842 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-sound
 1843 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-smartcard
 1850 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-wacom
 1854 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-a11y-settings
 1858 tty2      Sl+   0:03 /usr/lib/gnome-settings-daemon/gsd-clipboard
 1859 tty2      Sl+   0:21 /usr/lib/gnome-settings-daemon/gsd-color
 1860 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-datetime
 1861 tty2      Sl+   0:01 /usr/lib/gnome-settings-daemon/gsd-housekeeping
 1865 tty2      Sl+   0:03 /usr/lib/gnome-settings-daemon/gsd-keyboard
 1869 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-media-keys
 1871 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-mouse
 1916 tty2      Sl+   0:00 /usr/lib/gnome-disk-utility/gsd-disk-utility-notify
 1923 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-printer
 1924 tty2      Sl+   2:54 /usr/lib/vmware-tools/sbin64/vmtoolsd -n vmusr --blockFd 3
 1925 tty2      Sl+   0:07 nautilus-desktop
 2476 tty2      Sl+   0:04 update-notifier
 2478 tty2      Sll+  0:32 /usr/bin/gnome-software --gapplication-service
 2633 tty2      Sl+   0:00 /usr/lib/deja-dup/deja-dup-monitor
 33227 pts/0    Ss    0:00 bash
 33413 pts/0    S+    0:00 make test11
 33414 pts/0    S+    0:00 /bin/sh -c ./sdriver.pl -t trace11.txt -s ./tsh -a "-p"
 33415 pts/0    S+    0:00 /usr/bin/perl ./sdriver.pl -t trace11.txt -s ./tsh -a -p
 33416 pts/0    S+    0:00 ./tsh -p
 33421 pts/0    R     0:00 /bin/ps a

```

# 计算机系统实验报告

```

demerzel@demerzel-virtual-machine:~/cocode/lab7/shlab-handout-hit$ 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] (33428) terminated by signal 2
tsh> /bin/ps a
  PID TTY          STAT TIME   COMMAND
 1200 tty1      Ssl+  0:00 /usr/lib/gdm3/gdm-wayland-session gnome-session --autostart /usr/share/gdm/greeter/autostart
 1239 tty1      Sl+   0:00 /usr/lib/gnome-session/gnome-session-binary --autostart /usr/share/gdm/greeter/autostart
 1401 tty1      Sl+   0:27 /usr/bin/gnome-shell
 1424 tty1      Sl+   9:40 /usr/bin/Xwayland :1024 -rootless -terminate -accessx -core -listen 4 -listen 5 -displayfd 6
 1465 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-xsettings
 1468 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-a11y-settings
 1472 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-clipboard
 1473 tty1      Sl+   0:21 /usr/lib/gnome-settings-daemon/gsd-color
 1474 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-datetime
 1476 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-housekeeping
 1484 tty1      Sl+   0:02 /usr/lib/gnome-settings-daemon/gsd-keyboard
 1487 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-media-keys
 1488 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-mouse
 1489 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-power
 1492 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-print-notifications
 1493 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-rfkill
 1494 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-screensaver-proxy
 1499 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-sharing
 1503 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-smartcard
 1504 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-sound
 1510 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-wacom
 1592 tty2      Ssl+  0:00 /usr/lib/gdm3/gdm-x-session --run-script env GNOME_SHELL_SESSION_MODE=ubuntu gnome-session --session=ubuntu
 1594 tty2      Rl+   6:45 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/1000/gdm/Xauthority -background none -noreset -keeptty -verbose 3
 1603 tty2      Sl+   0:01 /usr/lib/gnome-session/gnome-session-binary --session=ubuntu
 1739 tty2      Sl+   12:49 /usr/bin/gnome-shell
 1820 tty2      Sl+   0:01 /usr/lib/gnome-settings-daemon/gsd-power
 1821 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-print-notifications
 1823 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-rfkill
 1824 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-screensaver-proxy
 1829 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-sharing
 1841 tty2      Sl+   0:01 /usr/lib/gnome-settings-daemon/gsd-xsettings
 1842 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-sound
 1843 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-smartcard
 1850 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-wacom
 1854 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-a11y-settings
 1858 tty2      Sl+   0:03 /usr/lib/gnome-settings-daemon/gsd-clipboard
 1859 tty2      Sl+   0:21 /usr/lib/gnome-settings-daemon/gsd-color
 1860 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-datetime
 1861 tty2      Sl+   0:01 /usr/lib/gnome-settings-daemon/gsd-housekeeping
 1865 tty2      Sl+   0:03 /usr/lib/gnome-settings-daemon/gsd-keyboard
 1869 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-media-keys
 1871 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-mouse
 1916 tty2      Sl+   0:00 /usr/lib/gnome-disk-utility/gsd-disk-utility-notify
 1923 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-printer
 1924 tty2      Sl+   2:55 /usr/lib/vmware-tools/sbin64/vmtoolsd -n vmusr --blockFd 3
 1925 tty2      Sl+   0:07 nautilus-desktop
 2476 tty2      Sl+   0:04 update-notifier
 2478 tty2      Sll+  0:32 /usr/bin/gnome-software --gapplication-service
 2633 tty2      Sl+   0:00 /usr/lib/deja-dup/deja-dup-monitor
 33227 pts/0    Ss    0:00 bash
 33423 pts/0    S+    0:00 make rtest11
 33424 pts/0    S+    0:00 /bin/sh -c ./sdriver.pl -t trace11.txt -s ./tshref -a "-p"
 33425 pts/0    S+    0:00 /usr/bin/perl ./sdriver.pl -t trace11.txt -s ./tshref -a -p
 33426 pts/0    S+    0:00 ./tshref -p
 33432 pts/0    R     0:00 /bin/ps a

```

tsh 测试结果		tshref 测试结果	
测试结论		相同（一张截不下，分两张截的）	



## 4.3.12 测试用例 trace12.txt 的输出截图（1 分）

```

demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ 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] (33440) terminated by signal 20
tsh> jobs
[1] (33440) Stopped ./mysplit 4
tsh> /bin/ps a
  PID TTY          STAT TIME  COMMAND
 1200 tty1      Ssl+  0:00 /usr/lib/gdm3/gdm-wayland-session gnome-session --autostart /usr/share/gdm/greeter/autostart
 1239 tty1      Sl+   0:00 /usr/lib/gnome-session/gnome-session-binary --autostart /usr/share/gdm/greeter/autostart
 1401 tty1      Sl+   0:27 /usr/bin/gnome-shell
 1424 tty1      Sl+   9:40 /usr/bin/Xwayland :1024 -rootless -terminate -accessx -core -listen 4 -listen 5 -displayfd 6
 1465 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-xsettings
 1468 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-a11y-settings
 1472 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-clipboard
 1473 tty1      Sl+   0:21 /usr/lib/gnome-settings-daemon/gsd-color
 1474 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-datetime
 1476 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-housekeeping
 1484 tty1      Sl+   0:02 /usr/lib/gnome-settings-daemon/gsd-keyboard
 1487 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-media-keys
 1488 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-mouse
 1489 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-power
 1492 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-print-notifications
 1493 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-rfkill
 1494 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-screensaver-proxy
 1499 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-sharing
 1503 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-smartcard
 1504 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-sound
 1510 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-wacom
 1592 tty2      Ssl+  0:00 /usr/lib/gdm3/gdm-x-session --run-script env GNOME_SHELL_SESSION_MODE=ubuntu gnome-session --session=ubuntu
 1594 tty2      Sl+   6:47 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/1000/gdm/Xauthority -background none -noreset -keeptty -verbose 3
 1603 tty2      Sl+   0:01 /usr/lib/gnome-session/gnome-session-binary --session=ubuntu
 1739 tty2      Sl+   12:52 /usr/bin/gnome-shell
 1820 tty2      Sl+   0:01 /usr/lib/gnome-settings-daemon/gsd-power
 1821 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-print-notifications
 1823 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-rfkill
 1824 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-screensaver-proxy
 1829 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-sharing
 1841 tty2      Sl+   0:01 /usr/lib/gnome-settings-daemon/gsd-xsettings
 1842 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-sound
 1843 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-smartcard
 1850 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-wacom
 1854 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-a11y-settings
 1858 tty2      Sl+   0:03 /usr/lib/gnome-settings-daemon/gsd-clipboard
 1859 tty2      Sl+   0:21 /usr/lib/gnome-settings-daemon/gsd-color
 1860 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-datetime
 1861 tty2      Sl+   0:01 /usr/lib/gnome-settings-daemon/gsd-housekeeping
 1865 tty2      Sl+   0:03 /usr/lib/gnome-settings-daemon/gsd-keyboard
 1869 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-media-keys
 1871 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-mouse
 1916 tty2      Sl+   0:00 /usr/lib/gnome-disk-utility/gsd-disk-utility-notify
 1923 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-printer
 1924 tty2      Sl+   2:55 /usr/lib/vmware-tools/sbin64/vmtoolsd -n vmusr --blockFd 3
 1925 tty2      Sl+   0:07 nautilus-desktop
 2476 tty2      Sl+   0:04 update-notifier
 2478 tty2      Sll+  0:32 /usr/bin/gnome-software --gapplication-service
 2633 tty2      Sl+   0:00 /usr/lib/deja-dup/deja-dup-monitor
 33227 pts/0    Ss    0:00 bash
 33435 pts/0    S+    0:00 make test12
 33436 pts/0    S+    0:00 /bin/sh -c ./sdriver.pl -t trace12.txt -s ./tsh -a "-p"
 33437 pts/0    S+    0:00 /usr/bin/perl ./sdriver.pl -t trace12.txt -s ./tsh -a -p
 33438 pts/0    S+    0:00 ./tsh -p
 33440 pts/0    T      0:00 ./mysplit 4
 33441 pts/0    T      0:00 ./mysplit 4
 33444 pts/0    R      0:00 /bin/ps a

```

## 计算机系统实验报告

```
demerzel@demerzel-virtual-machine:~/cocode/lab7/shlab-handout-hit$ 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] (33450) stopped by signal 20
tsh> jobs
[1] (33450) Stopped ./mysplit 4
tsh> /bin/ps a
  PID TTY          STAT       TIME COMMAND
 1200 tty1      Ssl+      0:00 /usr/lib/gdm3/gdm-wayland-session gnome-session --autostart /usr/share/gdm/greeter/autostart
 1239 tty1      Sl+       0:00 /usr/lib/gnome-session/gnome-session-binary --autostart /usr/share/gdm/greeter/autostart
 1401 tty1      Sl+       0:27 /usr/bin/gnome-shell
 1424 tty1      Sl+       9:40 /usr/bin/Xwayland :1024 -rootless -terminate -accessx -core -listen 4 -listen 5 -displayfd 6
 1465 tty1      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-xsettings
 1468 tty1      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-a11y-settings
 1472 tty1      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-clipboard
 1473 tty1      Sl+       0:21 /usr/lib/gnome-settings-daemon/gsd-color
 1474 tty1      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-datetime
 1476 tty1      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-housekeeping
 1484 tty1      Sl+       0:02 /usr/lib/gnome-settings-daemon/gsd-keyboard
 1487 tty1      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-media-keys
 1488 tty1      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-mouse
 1489 tty1      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-power
 1492 tty1      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-print-notifications
 1493 tty1      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-rfkill
 1494 tty1      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-screensaver-proxy
 1499 tty1      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-sharing
 1503 tty1      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-smartcard
 1504 tty1      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-sound
 1510 tty1      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-wacom
 1592 tty2      Ssl+      0:00 /usr/lib/gdm3/gdm-x-session --run-script env GNOME_SHELL_SESSION_MODE=ubuntu gnome-session --session=ubuntu
 1594 tty2      Sl+       6:47 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/1000/gdm/Xauthority -background none -noreset -keeptty -verbose 3
 1603 tty2      Sl+       0:01 /usr/lib/gnome-session/gnome-session-binary --session=ubuntu
 1739 tty2      Sl+      12:52 /usr/bin/gnome-shell
 1820 tty2      Sl+       0:01 /usr/lib/gnome-settings-daemon/gsd-power
 1821 tty2      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-print-notifications
 1823 tty2      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-rfkill
 1824 tty2      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-screensaver-proxy
 1829 tty2      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-sharing
 1841 tty2      Sl+       0:01 /usr/lib/gnome-settings-daemon/gsd-xsettings
 1842 tty2      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-sound
 1843 tty2      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-smartcard
 1850 tty2      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-wacom
 1854 tty2      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-a11y-settings
 1858 tty2      Sl+       0:03 /usr/lib/gnome-settings-daemon/gsd-clipboard
 1859 tty2      Sl+       0:21 /usr/lib/gnome-settings-daemon/gsd-color
 1860 tty2      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-datetime
 1861 tty2      Sl+       0:01 /usr/lib/gnome-settings-daemon/gsd-housekeeping
 1865 tty2      Sl+       0:03 /usr/lib/gnome-settings-daemon/gsd-keyboard
 1869 tty2      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-media-keys
 1871 tty2      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-mouse
 1916 tty2      Sl+       0:00 /usr/lib/gnome-disk-utility/gsd-disk-utility-notify
 1923 tty2      Sl+       0:00 /usr/lib/gnome-settings-daemon/gsd-printer
 1924 tty2      Sl+       2:55 /usr/lib/vmware-tools/sbin64/vmtoolsd -n vmusr --blockFd 3
 1925 tty2      Sl+       0:07 nautilus-desktop
 2476 tty2      Sl+       0:04 update-notifier
 2478 tty2      Sll+      0:32 /usr/bin/gnome-software --gapplication-service
 2633 tty2      Sl+       0:00 /usr/lib/deja-dup/deja-dup-monitor
 33227 pts/0    ss        0:00 bash
 33445 pts/0    S+        0:00 make rtest12
 33446 pts/0    S+        0:00 /bin/sh -c ./sdriver.pl -t trace12.txt -s ./tshref -a "-p"
 33447 pts/0    S+        0:00 /usr/bin/perl ./sdriver.pl -t trace12.txt -s ./tshref -a -p
 33448 pts/0    S+        0:00 ./tshref -p
 33450 pts/0    T         0:00 ./mysplit 4
 33451 pts/0    T         0:00 ./mysplit 4
 33454 pts/0    R         0:00 /bin/ps a
```

tsh 测试结果		tshref 测试结果	
测试结论		相同（一张没截下，分两张截的）	

## 4.3.13 测试用例 trace13.txt 的输出截图（1 分）

```

demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make test13
./sdriver.pl -t trace13.txt -s ./tsh -a "-p"
#
# trace13.txt - Restart every stopped process in process group
#
tsh> ./mysplit 4
Job [1] (33461) terminated by signal 20
tsh> jobs
[1] (33461) Stopped ./mysplit 4
tsh> /bin/ps a
  PID TTY          STAT TIME  COMMAND
 1200 tty1      Ssl+  0:00 /usr/lib/gdm3/gdm-wayland-session gnome-session --autostart /usr/share/gdm/greeter/autostart
 1239 tty1      Sl+   0:00 /usr/lib/gnome-session/gnome-session-binary --autostart /usr/share/gdm/greeter/autostart
 1401 tty1      Sl+   0:27 /usr/bin/gnome-shell
 1424 tty1      Sl+   9:41 /usr/bin/Xwayland :1024 -rootless -terminate -accessx -core -listen 4 -listen 5 -displayfd 6
 1465 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-xsettings
 1468 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-a11y-settings
 1472 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-clipboard
 1473 tty1      Sl+   0:21 /usr/lib/gnome-settings-daemon/gsd-color
 1474 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-datetime
 1476 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-housekeeping
 1484 tty1      Sl+   0:02 /usr/lib/gnome-settings-daemon/gsd-keyboard
 1487 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-media-keys
 1488 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-mouse
 1489 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-power
 1492 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-print-notifications
 1493 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-rfkill
 1494 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-screensaver-proxy
 1499 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-sharing
 1503 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-smartcard
 1504 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-sound
 1510 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-wacom
 1592 tty2      Ssl+  0:00 /usr/lib/gdm3/gdm-x-session --run-script env GNOME_SHELL_SESSION_MODE=ubuntu gnome-session --session=ubuntu
 1594 tty2      Sl+   6:48 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/1000/gdm/Xauthority -background none -noreset -keeptty -verbose 3
 1603 tty2      Sl+   0:01 /usr/lib/gnome-session/gnome-session-binary --session=ubuntu
 1739 tty2      Sl+   12:52 /usr/bin/gnome-shell
 1820 tty2      Sl+   0:01 /usr/lib/gnome-settings-daemon/gsd-power
 1821 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-print-notifications
 1823 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-rfkill
 1824 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-screensaver-proxy
 1829 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-sharing
 1841 tty2      Sl+   0:01 /usr/lib/gnome-settings-daemon/gsd-xsettings
 1842 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-sound
 1843 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-smartcard
 1850 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-wacom
 1854 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-a11y-settings
 1858 tty2      Sl+   0:03 /usr/lib/gnome-settings-daemon/gsd-clipboard
 1859 tty2      Sl+   0:21 /usr/lib/gnome-settings-daemon/gsd-color
 1860 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-datetime
 1861 tty2      Sl+   0:01 /usr/lib/gnome-settings-daemon/gsd-housekeeping
 1865 tty2      Sl+   0:03 /usr/lib/gnome-settings-daemon/gsd-keyboard
 1869 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-media-keys
 1871 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-mouse
 1916 tty2      Sl+   0:00 /usr/lib/gnome-disk-utility/gsd-disk-utility-notify
 1923 tty2      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-printer
 1924 tty2      Sl+   2:55 /usr/lib/vmware-tools/sbin64/vmtoolsd -n vmusr --blockFd 3
 1925 tty2      Sl+   0:07 nautilus-desktop
 2476 tty2      Sl+   0:04 update-notifier
 2478 tty2      Sll+  0:32 /usr/bin/gnome-software --gapplication-service
 2633 tty2      Sl+   0:00 /usr/lib/deja-dup/deja-dup-monitor
 33227 pts/0    Ss    0:00 bash
 33456 pts/0    S+    0:00 make test13
 33457 pts/0    S+    0:00 /bin/sh -c ./sdriver.pl -t trace13.txt -s ./tsh -a "-p"
 33458 pts/0    S+    0:00 /usr/bin/perl ./sdriver.pl -t trace13.txt -s ./tsh -a -p
 33459 pts/0    S+    0:00 ./tsh -p
 33461 pts/0    T      0:00 ./mysplit 4
 33462 pts/0    T      0:00 ./mysplit 4
 33465 pts/0    R      0:00 /bin/ps a
tsh> fg %1
make tsh> /bin/ps a
  PID TTY          STAT TIME  COMMAND
 1200 tty1      Ssl+  0:00 /usr/lib/gdm3/gdm-wayland-session gnome-session --autostart /usr/share/gdm/greeter/autostart
 1239 tty1      Sl+   0:00 /usr/lib/gnome-session/gnome-session-binary --autostart /usr/share/gdm/greeter/autostart
 1401 tty1      Sl+   0:27 /usr/bin/gnome-shell
 1424 tty1      Sl+   9:41 /usr/bin/Xwayland :1024 -rootless -terminate -accessx -core -listen 4 -listen 5 -displayfd 6
 1465 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-xsettings
 1468 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-a11y-settings
 1472 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-clipboard
 1473 tty1      Sl+   0:21 /usr/lib/gnome-settings-daemon/gsd-color
 1474 tty1      Sl+   0:00 /usr/lib/gnome-settings-daemon/gsd-datetime

```

## 计算机系统实验报告

```

1476 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-housekeeping
1484 tty1 Sl+ 0:02 /usr/lib/gnome-settings-daemon/gsd-keyboard
1487 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-media-keys
1488 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-mouse
1489 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-power
1492 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-print-notifications
1493 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-rfkill
1494 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-screensaver-proxy
1499 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-sharing
1503 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-smartcard
1504 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-sound
1510 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-wacom
1592 tty2 Ssl+ 0:00 /usr/lib/gdm3/gdm-x-session --run-script env GNOME_SHELL_SESSION_MODE=ubuntu gnome-session --session=ubuntu
1594 tty2 Sl+ 6:48 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/1000/gdm/Xauthority -background none -noreset -keeptty -verbose 3
1603 tty2 Sl+ 0:01 /usr/lib/gnome-session/gnome-session-binary --session=ubuntu
1739 tty2 Sl+ 12:52 /usr/bin/gnome-shell
1820 tty2 Sl+ 0:01 /usr/lib/gnome-settings-daemon/gsd-power
1821 tty2 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-print-notifications
1823 tty2 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-rfkill
1824 tty2 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-screensaver-proxy
1829 tty2 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-sharing
1841 tty2 Sl+ 0:01 /usr/lib/gnome-settings-daemon/gsd-xsettings
1842 tty2 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-sound
1843 tty2 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-smartcard
1850 tty2 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-wacom
1854 tty2 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-a11y-settings
1858 tty2 Sl+ 0:03 /usr/lib/gnome-settings-daemon/gsd-clipboard
1859 tty2 Sl+ 0:21 /usr/lib/gnome-settings-daemon/gsd-color
1860 tty2 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-datetime
1861 tty2 Sl+ 0:01 /usr/lib/gnome-settings-daemon/gsd-housekeeping
1865 tty2 Sl+ 0:03 /usr/lib/gnome-settings-daemon/gsd-keyboard
1869 tty2 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-media-keys
1871 tty2 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-mouse
1916 tty2 Sl+ 0:00 /usr/lib/gnome-disk-utility/gsd-disk-utility-notify
1923 tty2 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-printer
1924 tty2 Sl+ 2:55 /usr/lib/vmware-tools/sbin64/vmtoolsd -n vmusr --blockFd 3
1925 tty2 Sl+ 0:07 nautilus-desktop
2476 tty2 Sl+ 0:04 update-notifier
2478 tty2 Sll+ 0:32 /usr/bin/gnome-software --gapplication-service
2633 tty2 Sl+ 0:00 /usr/lib/deja-dup/deja-dup-monitor
33227 pts/0 Ss 0:00 bash
33456 pts/0 S+ 0:00 make test13
33457 pts/0 S+ 0:00 /bin/sh -c ./sdriver.pl -t trace13.txt -s ./tsh -a "-p"
33458 pts/0 S+ 0:00 /usr/bin/perl ./sdriver.pl -t trace13.txt -s ./tsh -a -p
33459 pts/0 S+ 0:00 ./tsh -p
33468 pts/0 R 0:00 /bin/ps a

```

```

demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make rtest13
./sdriver.pl -t trace13.txt -s ./tshref -a "-p"
#
# trace13.txt - Restart every stopped process in process group
#
tsh> ./mysplit 4
Job [1] (33474) stopped by signal 20
tsh> jobs
[1] (33474) Stopped ./mysplit 4
tsh> /bin/ps a
  PID TTY          STAT TIME  COMMAND
1200 tty1 Ssl+ 0:00 /usr/lib/gdm3/gdm-wayland-session gnome-session --autostart /usr/share/gdm/greeter/autostart
1239 tty1 Sl+ 0:00 /usr/lib/gnome-session/gnome-session-binary --autostart /usr/share/gdm/greeter/autostart
1401 tty1 Sl+ 0:27 /usr/bin/gnome-shell
1424 tty1 Sl+ 9:41 /usr/bin/Xwayland :1024 -rootless -terminate -accessx -core -listen 4 -listen 5 -displayfd 6
1465 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-xsettings
1468 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-a11y-settings
1472 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-clipboard
1473 tty1 Sl+ 0:21 /usr/lib/gnome-settings-daemon/gsd-color
1474 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-datetime
1476 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-housekeeping
1484 tty1 Sl+ 0:02 /usr/lib/gnome-settings-daemon/gsd-keyboard
1487 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-media-keys
1488 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-mouse
1489 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-power
1492 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-print-notifications
1493 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-rfkill
1494 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-screensaver-proxy
1499 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-sharing
1503 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-smartcard
1504 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-sound
1510 tty1 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-wacom
1592 tty2 Ssl+ 0:00 /usr/lib/gdm3/gdm-x-session --run-script env GNOME_SHELL_SESSION_MODE=ubuntu gnome-session --session=ubuntu
1594 tty2 Sl+ 6:48 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/1000/gdm/Xauthority -background none -noreset -keeptty -verbose 3
1603 tty2 Sl+ 0:01 /usr/lib/gnome-session/gnome-session-binary --session=ubuntu
1739 tty2 Sl+ 12:52 /usr/bin/gnome-shell
1820 tty2 Sl+ 0:01 /usr/lib/gnome-settings-daemon/gsd-power
1821 tty2 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-print-notifications
1823 tty2 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-rfkill
1824 tty2 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-screensaver-proxy
1829 tty2 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-sharing
1841 tty2 Sl+ 0:01 /usr/lib/gnome-settings-daemon/gsd-xsettings
1842 tty2 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-sound
1843 tty2 Sl+ 0:00 /usr/lib/gnome-settings-daemon/gsd-smartcard

```

## 计算机系统实验报告

```

1850 tty2 SL+ 0:00 /usr/lib/gnome-settings-daemon/gsd-wacom
1854 tty2 SL+ 0:00 /usr/lib/gnome-settings-daemon/gsd-a11y-settings
1858 tty2 SL+ 0:03 /usr/lib/gnome-settings-daemon/gsd-clipboard
1859 tty2 SL+ 0:21 /usr/lib/gnome-settings-daemon/gsd-color
1860 tty2 SL+ 0:00 /usr/lib/gnome-settings-daemon/gsd-datetime
1861 tty2 SL+ 0:01 /usr/lib/gnome-settings-daemon/gsd-housekeeping
1865 tty2 SL+ 0:03 /usr/lib/gnome-settings-daemon/gsd-keyboard
1869 tty2 SL+ 0:00 /usr/lib/gnome-settings-daemon/gsd-media-keys
1871 tty2 SL+ 0:00 /usr/lib/gnome-settings-daemon/gsd-mouse
1916 tty2 SL+ 0:00 /usr/lib/gnome-disk-utility/gsd-disk-utility-notify
1923 tty2 SL+ 0:00 /usr/lib/gnome-settings-daemon/gsd-printer
1924 tty2 SL+ 2:55 /usr/lib/vmware-tools/sbin64/vmtoolsd -n vmusr --blockFd 3
1925 tty2 SL+ 0:07 nautilus-desktop
2476 tty2 SL+ 0:04 update-notifier
2478 tty2 SL+ 0:32 /usr/bin/gnome-software --gapplication-service
2633 tty2 SL+ 0:00 /usr/lib/deja-dup/deja-dup-monitor
33227 pts/0 Ss 0:00 bash
33469 pts/0 S+ 0:00 make rtest13
33470 pts/0 S+ 0:00 /bin/sh -c ./sdriver.pl -t trace13.txt -s ./tshref -a "-p"
33471 pts/0 S+ 0:00 /usr/bin/perl ./sdriver.pl -t trace13.txt -s ./tshref -a -p
33472 pts/0 S+ 0:00 ./tshref -p
33474 pts/0 T 0:00 ./mysplit 4
33475 pts/0 T 0:00 ./mysplit 4
33478 pts/0 R 0:00 /bin/ps a
tsh> fg %1
tsh> /bin/ps a

```

PID	TTY	STAT	TIME	COMMAND
1200	tty1	Ssl+	0:00	/usr/lib/gdm3/gdm-wayland-session gnome-session --autostart /usr/share/gdm/greeter/autostart
1239	tty1	SL+	0:00	/usr/lib/gnome-session/gnome-session-binary --autostart /usr/share/gdm/greeter/autostart
1401	tty1	SL+	0:27	/usr/bin/gnome-shell
1424	tty1	SL+	9:41	/usr/bin/Xwayland :1024 -rootless -terminate -accessx -core -listen 4 -listen 5 -displayfd 6
1465	tty1	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-xsettings
1468	tty1	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-a11y-settings
1472	tty1	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-clipboard
1473	tty1	SL+	0:21	/usr/lib/gnome-settings-daemon/gsd-color
1474	tty1	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-datetime
1476	tty1	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-housekeeping
1484	tty1	SL+	0:02	/usr/lib/gnome-settings-daemon/gsd-keyboard
1487	tty1	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-media-keys
1488	tty1	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-mouse
1489	tty1	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-power
1492	tty1	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-print-notifications
1493	tty1	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-rfkill
1494	tty1	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-screensaver-proxy
1499	tty1	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-sharing
1503	tty1	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-smartcard
1504	tty1	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-sound
1510	tty1	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-wacom
1592	tty2	Ssl+	0:00	/usr/lib/gdm3/gdm-x-session --run-script env GNOME_SHELL_SESSION_MODE=ubuntu gnome-session --session=ubuntu
1594	tty2	SL+	6:48	/usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/1000/gdm/Xauthority -background none -noreset -keeptty -verbose 3
1603	tty2	SL+	0:01	/usr/lib/gnome-session/gnome-session-binary --session=ubuntu
1739	tty2	SL+	12:52	/usr/bin/gnome-shell
1820	tty2	SL+	0:01	/usr/lib/gnome-settings-daemon/gsd-power
1821	tty2	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-print-notifications
1823	tty2	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-rfkill
1824	tty2	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-screensaver-proxy
1829	tty2	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-sharing
1841	tty2	SL+	0:01	/usr/lib/gnome-settings-daemon/gsd-xsettings
1842	tty2	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-sound
1843	tty2	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-smartcard
1850	tty2	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-wacom
1854	tty2	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-a11y-settings
1858	tty2	SL+	0:03	/usr/lib/gnome-settings-daemon/gsd-clipboard
1859	tty2	SL+	0:21	/usr/lib/gnome-settings-daemon/gsd-color
1860	tty2	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-datetime
1861	tty2	SL+	0:01	/usr/lib/gnome-settings-daemon/gsd-housekeeping
1865	tty2	SL+	0:03	/usr/lib/gnome-settings-daemon/gsd-keyboard
1869	tty2	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-media-keys
1871	tty2	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-mouse
1916	tty2	SL+	0:00	/usr/lib/gnome-disk-utility/gsd-disk-utility-notify
1923	tty2	SL+	0:00	/usr/lib/gnome-settings-daemon/gsd-printer
1924	tty2	SL+	2:55	/usr/lib/vmware-tools/sbin64/vmtoolsd -n vmusr --blockFd 3
1925	tty2	SL+	0:07	nautilus-desktop
2476	tty2	SL+	0:04	update-notifier
2478	tty2	SL+	0:32	/usr/bin/gnome-software --gapplication-service
2633	tty2	SL+	0:00	/usr/lib/deja-dup/deja-dup-monitor
33227	pts/0	Ss	0:00	bash
33469	pts/0	S+	0:00	make rtest13
33470	pts/0	S+	0:00	/bin/sh -c ./sdriver.pl -t trace13.txt -s ./tshref -a "-p"
33471	pts/0	S+	0:00	/usr/bin/perl ./sdriver.pl -t trace13.txt -s ./tshref -a -p
33472	pts/0	S+	0:00	./tshref -p
33481	pts/0	R	0:00	/bin/ps a

tsh 测试结果		tshref 测试结果	
测试结论	相同（一张图截不下，分四张图截的）		



## 4.3.14 测试用例 trace14.txt 的输出截图（1 分）

```

demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make test14
./sdriver.pl -t trace14.txt -s ./tsh -a "-p"
#
# trace14.txt - Simple error handling
#
tsh> ./bogus
./bogus: Command not found
tsh> ./myspin 4 &
[1] (33489) ./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] (33489) terminated by signal 20
tsh> bg %2
%2: No such job
tsh> bg %1
[1] (33489) ./myspin 4 &
tsh> jobs
[1] (33489) Running ./myspin 4 &
demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make rtest14
./sdriver.pl -t trace14.txt -s ./tshref -a "-p"
#
# trace14.txt - Simple error handling
#
tsh> ./bogus
./bogus: Command not found
tsh> ./myspin 4 &
[1] (33597) ./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] (33597) stopped by signal 20
tsh> bg %2
%2: No such job
tsh> bg %1
[1] (33597) ./myspin 4 &
tsh> jobs
[1] (33597) Running ./myspin 4 &

```

tsh 测试结果		tshref 测试结果	
测试结论		相同	

## 4.3.15 测试用例 trace15.txt 的输出截图（1 分）

```

demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ make test15
./sdriver.pl -t trace15.txt -s ./tsh -a "-p"
#
# trace15.txt - Putting it all together
#
tsh> ./bogus
./bogus: Command not found
tsh> ./myspin 10
Job [1] (33528) terminated by signal 2
tsh> ./myspin 3 &
[1] (33530) ./myspin 3 &
tsh> ./myspin 4 &
[2] (33532) ./myspin 4 &
tsh> jobs
[1] (33530) Running ./myspin 3 &
[2] (33532) Running ./myspin 4 &
tsh> fg %1
Job [1] (33530) terminated by signal 20
tsh> jobs
[1] (33530) Stopped ./myspin 3 &
[2] (33532) Running ./myspin 4 &
tsh> bg %3
%3: No such job
tsh> bg %1
[1] (33530) ./myspin 3 &
tsh> jobs
[1] (33530) Running ./myspin 3 &
[2] (33532) Running ./myspin 4 &
tsh> fg %1
tsh> quit
demerzel@demerzel-virtual-machine:~/ccode/lab7/shlab-handout-hit$ 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] (33548) terminated by signal 2
tsh> ./myspin 3 &
[1] (33550) ./myspin 3 &
tsh> ./myspin 4 &
[2] (33552) ./myspin 4 &
tsh> jobs
[1] (33550) Running ./myspin 3 &
[2] (33552) Running ./myspin 4 &
tsh> fg %1
Job [1] (33550) stopped by signal 20
tsh> jobs
[1] (33550) Stopped ./myspin 3 &
[2] (33552) Running ./myspin 4 &
tsh> bg %3
%3: No such job
tsh> bg %1
[1] (33550) ./myspin 3 &
tsh> jobs
[1] (33550) Running ./myspin 3 &
[2] (33552) Running ./myspin 4 &
tsh> fg %1
tsh> quit

```

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： 15 。



## 第 4 章 总结

### 4.1 请总结本次实验的收获

第八章异常控制流的教材内容不多，但是概念多，前半部分还好，后半部分涉及进程控制时，调用的函数需要记忆的地方有很多，TinyShell 实验很好的帮助了我了解进程控制调用的过程。收获很多。

### 4.2 请给出对本次实验内容的建议

实验总体很好，就是截图太费劲了，15 个测试和样例，至少截图 30 张，有一张截不下的更闹心。

实验报告写的太麻烦了，字数真的是最多的一次。

## 参考文献

- [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 [1998-09-26]. <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[1998-09-23]. <http://www.sciencemag.org/cgi/collection/anatmorp>.