



哈尔滨工业大学
HARBIN INSTITUTE OF TECHNOLOGY

实验设计报告

开课学期： 2023 年春季
课程名称： 计算机系统
实验名称： Lab2 TinyShell
实验性质： 课内实验
实验时间： 地点：
学生班级： 计算机 6 班
学生学号： 200110619
学生姓名： 梁鑫嵘
评阅教师：
报告成绩：

实验与创新实践教育中心印制

2023 年 4 月

1. TinyShell 的设计与实现

1.1 设计

1.1.1 void eval(char *cmdline)函数

函数功能：根据命令行解析、执行程序逻辑

参 数：命令行字符串

处理流程：判断内置命令、fork、切换执行目标程序、管理jobs 结构、等待程序完成等

要点分析：

在 fork 之前需要屏蔽可能的信号：

```
sigprocmask(SIG_BLOCK, &sigset, NULL);
```

```
pid_t pid = fork();
```

```
if (pid == 0) {
```

```
// in child process
```

```
sigprocmask(SIG_UNBLOCK, &sigset, NULL);
```

在子进程内将进程移动到新的进程组

```
sigprocmask(SIG_UNBLOCK, &sigset, NULL);
```

```
setpgid(0, 0);
```

然后切换到目标程序进程

```
if (execve(argv[0], argv, environ) < 0) {
```

```
printf("%s: Command not found\n", argv[0]);
```

```
exit(0);
```

```
}
```

主进程内也需要取消屏蔽

```
sigprocmask(SIG_UNBLOCK, &sigset, NULL);
```

然后进行jobs 列表的管理

```
if (bg) {
```

```
addjob(jobs, pid, BG, cmdline);
```

```
printf("[%d] (%d) %s\n", pid2jid(pid), pid, cmdline);
```

```
} else {
```

```
addjob(jobs, pid, FG, cmdline);
```

```
}
```

之后等待前台进程结束

```
if (!bg) {
```

```
waitfg(pid);
```

```
}
```

1.1.2 int builtin_cmd(char **argv)函数

函数功能：判断命令参数是否为内置命令，如 bg、fg、jobs 等

参 数：解析后的参数列表

处理流程：逐个判断命令，如果是的话则调用对应函数，否则返回不是内置命令

要点分析：

边界处理

```
if (!argv || !argv[0]) return 0;
```

quit 表示退出

```
if (is_prefix(argv[0], "quit")) {
    Log("quit normally");
    exit(0);
}
```

jobs 表示列出当前管理的进程

```
} else if (is_prefix(argv[0], "jobs")) {
    Log("listing jobs");
    listjobs(jobs);
    return 1;
}
```

fg、bg 统一调用 do_bgfg

```
} else if (is_prefix(argv[0], "bg") || is_prefix(argv[0], "fg")) {
    do_bgfg(argv);
    return 1;
}
```

1.1.3 void do_bgfg(char **argv) 函数

函数功能：处理 bg、fg 内置命令

参 数：bg/fg 以及参数

处理流程：判断目标是 jid 还是 pid，然后向进程发送 SIGCONT 信号

要点分析：

发送信号函数

```
kill(-job->pid, SIGCONT);
```

如果是 fg，还需要等待回到前台执行

```
waitfg(job->pid);
```

1.1.4 void waitfg(pid_t pid) 函数

函数功能：等待 pid 进程变为前台

参 数：pid

处理流程：获得 job 信息，然后循环等待

要点分析：

获得 job 结构体信息

```
struct job_t *job = getjobpid(jobs, pid);  
if (!job) return;
```

等待到达前台

```
while (job->state == FG) {  
    // Log("waiting for foreground job: %d", pid);  
    sleep(1);  
}
```

这里可以 sleep 节省 cpu，如果是不保证内存原子性读的系统可能还需要信号量等保持数据原子同步读

1.1.5 void sigchld_handler(int sig) 函数

函数功能：处理子进程结束传来的 SIGCHLD 信号

参 数：信号 id，即 SIGCHLD

处理流程：如下

要点分析：

获取触发 SIGCHLD 的子进程信息。由于可能多个子进程触发，所以 WNOHANG 以快速处理不等待。

```
while ((pid = waitpid(-1, &status, WNOHANG | WUNTRACED)) > 0) {
```

使用 WIFEXITED(status) 获取进程是否是正常结束

使用 WIFSIGNALED 获取进程是否被信号所 terminated

使用 WIFSTOPPED 获取进程是否被信号强制终止，终止了则设置 state 为 ST

```
getjobpid(jobs, pid)->state = ST;
```

1.2 程序实现（无需贴图，只需提交 tsh.c 文件即可）

重点检查 tsh.c 的代码风格：

- (1) 用较好的代码注释说明
- (2) 检查每个系统调用的返回值

2. TinyShell 测试

2.1 测试方法

针对 tsh 和参考 shell 程序 tshref, 完成测试项目 trace01-trace16 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), 填写完成 2.3 节的相应表格。

2.2 测试结果评价

tsh 与 tshref 的输出在以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 mysplrit 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。

2.3 自测试结果

说明:

实验中本人修改了 Makefile, 通过 make dtestXX 的方式进行对比测试, 先运行 tshref 再运行 tsh, 进而可以比对出差异。以下输出均可判定为相同。

Makefile:

Makefile for the CS:APP Shell Lab

```

TEAM = NOBODY
VERSION = 1
HANDINDIR = /afs/cs/academic/class/15213-f02/L5/handin
DRIVER = ./sdriver.pl
TSH ?= ./tsh
TSH_DBG ?= $(TSH)-dbg
TSHREF = ./tshref
TSHARGS = "-p"
CC = gcc
CFLAGS = -Wall -O2
FILES = $(TSH) $(TSH_DBG) ./myspin ./mysplit ./mystop ./myint ./myecho

CFLAGS += -I./include

all: $(FILES)

$(TSH): tsh.c
$(CC) $(CFLAGS) -DUSE_TSH=1 -o $@ $< csapp.c

$(TSH_DBG): tsh.c
$(CC) $(CFLAGS) -DUSE_TSH=1 -DCONFIG_PRINT_LOG=1 -o $@ $< csapp.c

# all files
%: %.c
$(CC) $(CFLAGS) -o $@ $< csapp.c

#####
# Handin your work
#####
handin:
cp tsh.c $(HANDINDIR)/$(TEAM)-$(VERSION)-tsh.c

#####
# Regression tests
#####

# Run tests using the student's shell program
test%: $(FILES)
$(DRIVER) -t trace$.txt -s $(TSH) -a $(TSHARGS)

# Run the tests using the reference shell program
rtest%: $(FILES)
$(DRIVER) -t trace$.txt -s $(TSHREF) -a $(TSHARGS)

```

```

# difftest
dtest%: $(FILES)
$(DRIVER) -t trace$.txt -s $(TSHREF) -a $(TSHARGS)
$(DRIVER) -t trace$.txt -s $(TSH) -a $(TSHARGS)

# difftest debug
ddtest%: $(FILES)
$(DRIVER) -t trace$.txt -s $(TSHREF) -a $(TSHARGS)
$(DRIVER) -t trace$.txt -s $(TSH_DBG) -a $(TSHARGS)

run: $(TSH_DBG)
$(TSH_DBG)

# clean up
clean:
rm -f $(FILES) *.o *~

.PHONY: all clean run handin test% rtest% dtest%

```

2.3.1 测试用例 trace01.txt 的输出截图

tsh 测试结果	tshref 测试结果
<pre> test_19@comp1:~/csapp/shlab-handout\$ make test01 ./sdriver.pl -t trace01.txt -s ./tsh -a "-p" # # trace01.txt - Properly terminate on EOF. # test_19@comp1:~/csapp/shlab-handout\$ </pre>	<pre> test_19@comp1:~/csapp/shlab-handout\$ make rtest01 ./sdriver.pl -t trace01.txt -s ./tshref -a "-p" # # trace01.txt - Properly terminate on EOF. # test_19@comp1:~/csapp/shlab-handout\$ </pre>
测试结论	相同

2.3.2 测试用例 trace02.txt 的输出截图

```

● → chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) × make dtest02
./sdriver.pl -t trace02.txt -s ./tshref -a "-p"
#
# trace02.txt - Process builtin quit command.
#
./sdriver.pl -t trace02.txt -s ./tsh -a "-p"
#
# trace02.txt - Process builtin quit command.
#
○ → chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) ×

```

master* 10 0 Connect 2 hrs 28 mins Git Graph Chinese (Simplified)

2.3.3 测试用例 trace03.txt 的输出截图

```

● → chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) × make dtest03
./sdriver.pl -t trace03.txt -s ./tshref -a "-p"
#
# trace03.txt - Run a foreground job.
#
tsh> quit
./sdriver.pl -t trace03.txt -s ./tsh -a "-p"
#
# trace03.txt - Run a foreground job.
#
tsh> quit

```

2.3.4 测试用例 trace04.txt 的输出截图

```

● → chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) × make dtest04
./sdriver.pl -t trace04.txt -s ./tshref -a "-p"
#
# trace04.txt - Run a background job.
#
tsh> ./myspin 1 &
[1] (2701261) ./myspin 1 &
./sdriver.pl -t trace04.txt -s ./tsh -a "-p"
#
# trace04.txt - Run a background job.
#
tsh> ./myspin 1 &
[1] (2701272) ./myspin 1 &

```


2.3.5 测试用例 trace05.txt 的输出截图

```

● → chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) ✕ make dtest05
./sdriver.pl -t trace05.txt -s ./tshref -a "-p"
#
# trace05.txt - Process jobs builtin command.
#
tsh> ./myspin 2 &
[1] (2701399) ./myspin 2 &
tsh> ./myspin 3 &
[2] (2701401) ./myspin 3 &
tsh> jobs
[1] (2701399) Running ./myspin 2 &
[2] (2701401) Running ./myspin 3 &
./sdriver.pl -t trace05.txt -s ./tsh -a "-p"
#
# trace05.txt - Process jobs builtin command.
#
tsh> ./myspin 2 &
[1] (2701424) ./myspin 2 &

tsh> ./myspin 3 &
[2] (2701426) ./myspin 3 &

tsh> jobs
[1] (2701424) Running ./myspin 2 &
[2] (2701426) Running ./myspin 3 &

```

2.3.6 测试用例 trace06.txt 的输出截图

```

[1] (2701426) Running ./myspin 3 &
● → chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) ✕ make dtest06
./sdriver.pl -t trace06.txt -s ./tshref -a "-p"
#
# trace06.txt - Forward SIGINT to foreground job.
#
tsh> ./myspin 4
Job [1] (2701572) terminated by signal 2
./sdriver.pl -t trace06.txt -s ./tsh -a "-p"
#
# trace06.txt - Forward SIGINT to foreground job.
#
tsh> ./myspin 4
Job [1] (2701595) terminated by signal 2

```

2.3.7 测试用例 trace07.txt 的输出截图

```

chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) ✕ make dtest07
gcc -Wall -O2 -I./include -DUSE_TSH=1 -o tsh tsh.c csapp.c
gcc -Wall -O2 -I./include -DUSE_TSH=1 -DCONFIG_PRINT_LOG=1 -o tsh-dbg tsh.c csapp.c
./sdriver.pl -t trace07.txt -s ./tshref -a "-p"
#
# trace07.txt - Forward SIGINT only to foreground job.
#
tsh> ./myspin 4 &
[1] (2702033) ./myspin 4 &
tsh> ./myspin 5
Job [2] (2702035) terminated by signal 2
tsh> jobs
[1] (2702033) Running ./myspin 4 &
./sdriver.pl -t trace07.txt -s ./tsh -a "-p"
#
# trace07.txt - Forward SIGINT only to foreground job.
#
tsh> ./myspin 4 &
[1] (2702086) ./myspin 4 &
tsh> ./myspin 5
Job [2] (2702088) terminated by signal 2
tsh> jobs
[1] (2702086) Running ./myspin 4 &

```

2.3.8 测试用例 trace08.txt 的输出截图

```

chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) ✕ make dtest08
./sdriver.pl -t trace08.txt -s ./tshref -a "-p"
#
# trace08.txt - Forward SIGTSTP only to foreground job.
#
tsh> ./myspin 4 &
[1] (2702231) ./myspin 4 &
tsh> ./myspin 5
Job [2] (2702233) stopped by signal 20
tsh> jobs
[1] (2702231) Running ./myspin 4 &
[2] (2702233) Stopped ./myspin 5
./sdriver.pl -t trace08.txt -s ./tsh -a "-p"
#
# trace08.txt - Forward SIGTSTP only to foreground job.
#
tsh> ./myspin 4 &
[1] (2702264) ./myspin 4 &
tsh> ./myspin 5
Job [2] (2702266) stopped by signal 20
tsh> jobs
[1] (2702264) Running ./myspin 4 &
[2] (2702266) Stopped ./myspin 5

```

2.3.9 测试用例 trace09.txt 的输出截图

```

[2] (2702418) stopped ./myspin 5
● → chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) ✕ make dtest09
./sdriver.pl -t trace09.txt -s ./tshref -a "-p"
#
# trace09.txt - Process bg builtin command
#
tsh> ./myspin 4 &
[1] (2702416) ./myspin 4 &
tsh> ./myspin 5
Job [2] (2702418) stopped by signal 20
tsh> jobs
[1] (2702416) Running ./myspin 4 &
[2] (2702418) Stopped ./myspin 5
tsh> bg %2
[2] (2702418) ./myspin 5
tsh> jobs
[1] (2702416) Running ./myspin 4 &
[2] (2702418) Running ./myspin 5
./sdriver.pl -t trace09.txt -s ./tsh -a "-p"
#
# trace09.txt - Process bg builtin command
#
tsh> ./myspin 4 &
[1] (2702451) ./myspin 4 &
tsh> ./myspin 5
Job [2] (2702453) stopped by signal 20
tsh> jobs
[1] (2702451) Running ./myspin 4 &
[2] (2702453) Stopped ./myspin 5
tsh> bg %2
[2] (2702453) ./myspin 5
tsh> jobs
[1] (2702451) Running ./myspin 4 &
[2] (2702453) Running ./myspin 5

```

2.3.10 测试用例 trace10.txt 的输出截图

```
● ➔ chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) ✕ make dtest10
./sdriver.pl -t trace10.txt -s ./tshref -a "-p"
#
# trace10.txt - Process fg builtin command.
#
tsh> ./myspin 4 &
[1] (2702612) ./myspin 4 &
tsh> fg %1
Job [1] (2702612) stopped by signal 20
tsh> jobs
[1] (2702612) Stopped ./myspin 4 &
tsh> fg %1
tsh> jobs
./sdriver.pl -t trace10.txt -s ./tsh -a "-p"
#
# trace10.txt - Process fg builtin command.
#
tsh> ./myspin 4 &
[1] (2702642) ./myspin 4 &
tsh> fg %1
Job [1] (2702642) stopped by signal 20
tsh> jobs
[1] (2702642) Stopped ./myspin 4 &
tsh> fg %1
tsh> jobs
○ ➔ chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) ✕
```

2.3.11 测试用例 trace11.txt 的输出截图

```

chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) × make dtest11
./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] (2702786) terminated by signal 2
tsh> /bin/ps a
      PID TTY          STAT TIME  COMMAND
    3566 tty1      Ssl+  27:22 /usr/lib/Xorg -nolisten tcp -background none -seat seat0 vt1 -auth /var/run/
sddm/{45dc833f-89dc-4aa7-9cb2-d588ca540bbc} -noreset -displayfd 17
    85025 pts/4      Ss+    0:01 /bin/zsh
   2638787 pts/1      Ss     0:00 /usr/bin/zsh -i
   2667867 pts/3      Ss+    0:00 /bin/zsh
   2670101 pts/6      Ss+    0:00 /bin/zsh
   2695024 pts/2      Ss     0:00 /bin/zsh
   2695378 pts/2      S       0:00 sudo su
   2695379 pts/2      S       0:00 su
   2695380 pts/2      S+     0:00 zsh
   2699823 pts/1      R       7:03 ./tsh -p
   2699990 pts/1      R       6:51 ./tsh -p
   2700118 pts/1      R       6:16 ./tsh -p
   2702782 pts/1      S+     0:00 make dtest11
   2702783 pts/1      S+     0:00 /usr/bin/perl ./sdriver.pl -t trace11.txt -s ./tshref -a -p
   2702784 pts/1      S+     0:00 ./tshref -p
   2702803 pts/1      R       0:00 /bin/ps a
./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] (2702818) terminated by signal 2
tsh> /bin/ps a
      PID TTY          STAT TIME  COMMAND
    3566 tty1      Ssl+  27:22 /usr/lib/Xorg -nolisten tcp -background none -seat seat0 vt1 -auth /var/run/
sddm/{45dc833f-89dc-4aa7-9cb2-d588ca540bbc} -noreset -displayfd 17
    85025 pts/4      Ss+    0:01 /bin/zsh
   2638787 pts/1      Ss     0:00 /usr/bin/zsh -i
   2667867 pts/3      Ss+    0:00 /bin/zsh
   2670101 pts/6      Ss+    0:00 /bin/zsh
   2695024 pts/2      Ss     0:00 /bin/zsh
   2695378 pts/2      S       0:00 sudo su
   2695379 pts/2      S       0:00 su
   2695380 pts/2      S+     0:00 zsh
   2699823 pts/1      R       7:05 ./tsh -p
   2699990 pts/1      R       6:53 ./tsh -p
   2700118 pts/1      R       6:18 ./tsh -p
   2702782 pts/1      S+     0:00 make dtest11
   2702815 pts/1      S+     0:00 /usr/bin/perl ./sdriver.pl -t trace11.txt -s ./tsh -a -p
   2702816 pts/1      S+     0:00 ./tsh -p
   2702832 pts/1      R       0:00 /bin/ps a
chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) ×

```

2.3.12 测试用例 trace12.txt 的输出截图

```

● chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) × make dtest12
./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] (2702977) stopped by signal 20
tsh> jobs
[1] (2702977) Stopped ./mysplit 4
tsh> /bin/ps a
  PID TTY          STAT TIME  COMMAND
 3566 tty1      Ss1+  27:25 /usr/lib/Xorg -nolisten tcp -background none -seat seat0 vt1 -auth /var/run/sddm/{45dc833f-89dc-4aa7-9cb2-d588ca540bbc} -noreset -displayfd 17
 85025 pts/4      Ss+   0:01 /bin/zsh
2638787 pts/1      Ss    0:00 /usr/bin/zsh -i
2667867 pts/3      Ss+   0:00 /bin/zsh
2670101 pts/6      Ss+   0:00 /bin/zsh
2695024 pts/2      Ss    0:00 /bin/zsh
2695378 pts/2      S     0:00 sudo su
2695379 pts/2      S     0:00 su
2695380 pts/2      S+    0:00 zsh
2699823 pts/1      R     7:22 ./tsh -p
2699990 pts/1      R     7:10 ./tsh -p
2700118 pts/1      R     6:35 ./tsh -p
2702973 pts/1      S+    0:00 make dtest12
2702974 pts/1      S+    0:00 /usr/bin/perl ./sdriver.pl -t trace12.txt -s ./tshref -a -p
2702975 pts/1      S+    0:00 ./tshref -p
2702977 pts/1      T     0:00 ./mysplit 4
2702978 pts/1      T     0:00 ./mysplit 4
2702992 pts/1      R     0:00 /bin/ps a
./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] (2702997) stopped by signal 20
tsh> jobs
[1] (2702997) Stopped ./mysplit 4
tsh> /bin/ps a
  PID TTY          STAT TIME  COMMAND
 3566 tty1      Ss1+  27:25 /usr/lib/Xorg -nolisten tcp -background none -seat seat0 vt1 -auth /var/run/sddm/{45dc833f-89dc-4aa7-9cb2-d588ca540bbc} -noreset -displayfd 17
 85025 pts/4      Ss+   0:01 /bin/zsh
2638787 pts/1      Ss    0:00 /usr/bin/zsh -i
2667867 pts/3      Ss+   0:00 /bin/zsh
2670101 pts/6      Ss+   0:00 /bin/zsh
2695024 pts/2      Ss    0:00 /bin/zsh
2695378 pts/2      S     0:00 sudo su
2695379 pts/2      S     0:00 su
2695380 pts/2      S+    0:00 zsh
2699823 pts/1      R     7:24 ./tsh -p
2699990 pts/1      R     7:12 ./tsh -p
2700118 pts/1      R     6:37 ./tsh -p
2702973 pts/1      S+    0:00 make dtest12
2702993 pts/1      S+    0:00 /usr/bin/perl ./sdriver.pl -t trace12.txt -s ./tsh -a -p
2702994 pts/1      S+    0:00 ./tsh -p
2702997 pts/1      T     0:00 ./mysplit 4
2702998 pts/1      T     0:00 ./mysplit 4
2703020 pts/1      R     0:00 /bin/ps a
● chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) ×

```

2.3.13 测试用例 trace13.txt 的输出截图

```

chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) × make dtest13
./sdriver.pl -t trace13.txt -s ./tshref -a "-p"
#
# trace13.txt - Restart every stopped process in process group
#
tsh> ./mysplit 4
Job [1] (2703169) stopped by signal 20
tsh> jobs
[1] (2703169) Stopped ./mysplit 4
tsh> /bin/ps a
  PID TTY          STAT       TIME COMMAND
    3566 tty1      Ssl+    27:27 /usr/lib/Xorg -nolisten tcp -background none -seat seat0 vt1 -auth /var/run/sddm/{45dc833f-89dc-4aa7-9cb2-d588ca540bbc} -noreset -displayfd 17
   85025 pts/4      Ss+      0:01 /bin/zsh
  2638787 pts/1      Ss        0:00 /usr/bin/zsh -i
  2667867 pts/3      Ss+      0:00 /bin/zsh
  2670101 pts/6      Ss+      0:00 /bin/zsh
  2695024 pts/2      Ss        0:00 /bin/zsh
  2695378 pts/2      S          0:00 sudo su
  2695380 pts/2      S+        0:00 zsh
  2699823 pts/1      R          7:44 ./tsh -p
  2699990 pts/1      R          7:32 ./tsh -p
  2700118 pts/1      R          6:57 ./tsh -p
  2703165 pts/1      S+        0:00 make dtest13
  2703166 pts/1      S+        0:00 /usr/bin/perl ./sdriver.pl -t trace13.txt -s ./tshref -a -p
  2703167 pts/1      S+        0:00 ./tshref -p
  2703169 pts/1      T          0:00 ./mysplit 4
  2703170 pts/1      T          0:00 ./mysplit 4
  2703185 pts/1      R          0:00 /bin/ps a
tsh> fg %1
tsh> /bin/ps a
  PID TTY          STAT       TIME COMMAND
    3566 tty1      Ssl+    27:27 /usr/lib/Xorg -nolisten tcp -background none -seat seat0 vt1 -auth /var/run/sddm/{45dc833f-89dc-4aa7-9cb2-d588ca540bbc} -noreset -displayfd 17
   85025 pts/4      Ss+      0:01 /bin/zsh
  2638787 pts/1      Ss        0:00 /usr/bin/zsh -i
  2667867 pts/3      Ss+      0:00 /bin/zsh
  2670101 pts/6      Ss+      0:00 /bin/zsh
  2695024 pts/2      Ss        0:00 /bin/zsh
  2695378 pts/2      S          0:00 sudo su
  2695380 pts/2      S+        0:00 su
  2699823 pts/1      R          7:46 ./tsh -p
  2699990 pts/1      R          7:34 ./tsh -p
  2700118 pts/1      R          6:59 ./tsh -p
  2703165 pts/1      S+        0:00 make dtest13
  2703166 pts/1      S+        0:00 /usr/bin/perl ./sdriver.pl -t trace13.txt -s ./tshref -a -p
  2703167 pts/1      S+        0:00 ./tshref -p
  2703200 pts/1      R          0:00 /bin/ps a
./sdriver.pl -t trace13.txt -s ./tsh -a "-p"
#
# trace13.txt - Restart every stopped process in process group
#
tsh> ./mysplit 4
Job [1] (2703204) stopped by signal 20
tsh> jobs
[1] (2703204) Stopped ./mysplit 4
tsh> /bin/ps a
  PID TTY          STAT       TIME COMMAND
    3566 tty1      Ssl+    27:27 /usr/lib/Xorg -nolisten tcp -background none -seat seat0 vt1 -auth /var/run/sddm/{45dc833f-89dc-4aa7-9cb2-d588ca540bbc} -noreset -displayfd 17
   85025 pts/4      Ss+      0:01 /bin/zsh
  2638787 pts/1      Ss        0:00 /usr/bin/zsh -i
  2667867 pts/3      Ss+      0:00 /bin/zsh
  2670101 pts/6      Ss+      0:00 /bin/zsh
  2695024 pts/2      Ss        0:00 /bin/zsh
  2695378 pts/2      S          0:00 sudo su
  2695380 pts/2      S+        0:00 su
  2699823 pts/1      R          7:48 ./tsh -p
  2699990 pts/1      R          7:36 ./tsh -p
  2700118 pts/1      R          7:01 ./tsh -p
  2703165 pts/1      S+        0:00 make dtest13
  2703201 pts/1      S+        0:00 /usr/bin/perl ./sdriver.pl -t trace13.txt -s ./tsh -a -p
  2703202 pts/1      S+        0:00 ./tsh -p
  2703204 pts/1      T          0:00 ./mysplit 4
  2703205 pts/1      T          0:00 ./mysplit 4
  2703219 pts/1      R          0:00 /bin/ps a
tsh> fg %1
tsh> /bin/ps a
  PID TTY          STAT       TIME COMMAND
    3566 tty1      Ssl+    27:27 /usr/lib/Xorg -nolisten tcp -background none -seat seat0 vt1 -auth /var/run/sddm/{45dc833f-89dc-4aa7-9cb2-d588ca540bbc} -noreset -displayfd 17
   85025 pts/4      Ss+      0:01 /bin/zsh
  2638787 pts/1      Ss        0:00 /usr/bin/zsh -i
  2667867 pts/3      Ss+      0:00 /bin/zsh
  2670101 pts/6      Ss+      0:00 /bin/zsh
  2695024 pts/2      Ss        0:00 /bin/zsh
  2695378 pts/2      S          0:00 sudo su
  2695380 pts/2      S+        0:00 su
  2699823 pts/1      R          7:50 ./tsh -p
  2699990 pts/1      R          7:38 ./tsh -p
  2700118 pts/1      R          7:03 ./tsh -p
  2703165 pts/1      S+        0:00 make dtest13
  2703201 pts/1      S+        0:00 /usr/bin/perl ./sdriver.pl -t trace13.txt -s ./tsh -a -p
  2703202 pts/1      S+        0:00 ./tsh -p
  2703235 pts/1      R          0:00 /bin/ps a

```

2.3.14 测试用例 trace14.txt 的输出截图

```

2703551 pts/1      K            0:00 /bin/ps -a
chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) × make dtest14
./sdriver.pl -t trace14.txt -s ./tshref -a "-p"
#
# trace14.txt - Simple error handling
#
tsh> ./bogus
./bogus: Command not found
tsh> ./myspin 4 &
[1] (2703513) ./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] (2703513) stopped by signal 20
tsh> bg %2
%2: No such job
tsh> bg %1
[1] (2703513) ./myspin 4 &
tsh> jobs
[1] (2703513) Running ./myspin 4 &
./sdriver.pl -t trace14.txt -s ./tsh -a "-p"
#
# trace14.txt - Simple error handling
#
tsh> ./bogus
./bogus: Command not found
tsh> ./myspin 4 &
[1] (2703551) ./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] (2703551) stopped by signal 20
tsh> bg %2
%2: No such job
tsh> bg %1
[1] (2703551) ./myspin 4 &
tsh> jobs
[1] (2703551) Running ./myspin 4 &

```


2.3.15 测试用例 trace15.txt 的输出截图

```
[1] (2703551) Running ./myspin 4 &
● ➤ chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) ✕ make dtest15
./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] (2703702) terminated by signal 2
tsh> ./myspin 3 &
[1] (2703715) ./myspin 3 &
tsh> ./myspin 4 &
[2] (2703718) ./myspin 4 &
tsh> jobs
[1] (2703715) Running ./myspin 3 &
[2] (2703718) Running ./myspin 4 &
tsh> fg %1
Job [1] (2703715) stopped by signal 20
tsh> jobs
[1] (2703715) Stopped ./myspin 3 &
[2] (2703718) Running ./myspin 4 &
tsh> bg %3
%3: No such job
tsh> bg %1
[1] (2703715) ./myspin 3 &
tsh> jobs
[1] (2703715) Running ./myspin 3 &
[2] (2703718) Running ./myspin 4 &
tsh> fg %1
tsh> quit
./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] (2703753) terminated by signal 2
tsh> ./myspin 3 &
[1] (2703766) ./myspin 3 &
tsh> ./myspin 4 &
[2] (2703768) ./myspin 4 &
tsh> jobs
[1] (2703766) Running ./myspin 3 &
[2] (2703768) Running ./myspin 4 &
tsh> fg %1
Job [1] (2703766) stopped by signal 20
tsh> jobs
[1] (2703766) Stopped ./myspin 3 &
[2] (2703768) Running ./myspin 4 &
tsh> bg %3
%3: No such job
tsh> bg %1
[1] (2703766) ./myspin 3 &
tsh> jobs
[1] (2703766) Running ./myspin 3 &
[2] (2703768) Running ./myspin 4 &
tsh> fg %1
tsh> quit
○ ➤ chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) ✕
```

2.3.16 测试用例 trace16.txt 的输出截图

```
tsn> quit
● → chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) × make dtest16
./sdriver.pl -t trace16.txt -s ./tshref -a "-p"
#
# trace16.txt - Tests whether the shell can handle SIGTSTP and SIGINT
#      signals that come from other processes instead of the terminal.
#
tsh> ./mystop 2
Job [1] (2703931) stopped by signal 20
tsh> jobs
[1] (2703931) Stopped ./mystop 2
tsh> ./myint 2
Job [2] (2703956) terminated by signal 2
./sdriver.pl -t trace16.txt -s ./tsh -a "-p"
#
# trace16.txt - Tests whether the shell can handle SIGTSTP and SIGINT
#      signals that come from other processes instead of the terminal.
#
tsh> ./mystop 2
Job [1] (2703971) stopped by signal 20
tsh> jobs
[1] (2703971) Stopped ./mystop 2
tsh> ./myint 2
Job [2] (2703984) terminated by signal 2
○ → chiro@chiro-pc ~/programs/comp3052-lab/lab2 git:(master) ×
```

3. 实验中遇到的问题及解决方法

(包括设计过程中的错误及测试过程中遇到的问题)

1. 我想在编写程序的过程中输出调试信息，但是不想影响文本比对：
 - a) 导入了我自己写的一个 C 语言 Logger 库
 - b) 添加一个 target: tsh-dbg，为其添加额外的 define，从而实现输出 logger
 - c) 原 tsh 不受影响，在不定义对应 flag 情况下不会有调试输出
2. Makefile 好麻烦
 - a) 改写为了上面的样式
 - b) 通过匹配减少了代码冗余
 - c) 添加了两种 diff test 模式
3. Signal 难以理解
 - a) 看书。

4. 请总结实验的收获，并给出对实验内容的建议

注：本章为酌情加分项。

好。选用了优秀的国外实验。

1. buflab

能不能让这个溢出过程有一个可视化的过程？看 gdb 很考验想象力的。

2. tinysHELL

下次可以加上自动评测或自动检查，手动比对很难受。