

**实验报告**

**实 验（七）**

题 目 TinyShell

微壳

专 业 计算机大类

学　　 号 1191000606

班　　 级 1903003

学 生 陈一帆

指 导 教 师 郑贵滨

实 验 地 点 G709

实 验 日 期 2021.6.4

**计算机科学与技术学院**

**目 录**

[第1章 实验基本信息 - 4 -](#_Toc26657523)

[1.1 实验目的 - 4 -](#_Toc26657524)

[1.2 实验环境与工具 - 4 -](#_Toc26657525)

[1.2.1 硬件环境 - 4 -](#_Toc26657526)

[1.2.2 软件环境 - 4 -](#_Toc26657527)

[1.2.3 开发工具 - 4 -](#_Toc26657528)

[1.3 实验预习 - 4 -](#_Toc26657529)

[第2章 实验预习 - 6 -](#_Toc26657530)

[2.1 进程的概念、创建和回收方法（5分） - 6 -](#_Toc26657531)

[2.2信号的机制、种类（5分） - 6 -](#_Toc26657532)

[2.3 信号的发送方法、阻塞方法、处理程序的设置方法（5分） - 6 -](#_Toc26657533)

[2.4 什么是shell，功能和处理流程（5分） - 6 -](#_Toc26657534)

[第3章 TinyShell的设计与实现 - 7 -](#_Toc26657535)

[3.1.1 void eval(char \*cmdline)函数（10分） - 7 -](#_Toc26657536)

[3. 1.2 int builtin\_cmd(char \*\*argv)函数（5分） - 7 -](#_Toc26657537)

[3. 1.3 void do\_bgfg(char \*\*argv) 函数（5分） - 7 -](#_Toc26657538)

[3. 1.4 void waitfg(pid\_t pid) 函数（5分） - 7 -](#_Toc26657539)

[3. 1.5 void sigchld\_handler(int sig) 函数（10分） - 8 -](#_Toc26657540)

[第4章 TinyShell测试 - 10 -](#_Toc26657541)

[4.1 测试方法 - 10 -](#_Toc26657542)

[4.2 测试结果评价 - 10 -](#_Toc26657543)

[4.3 自测试结果 - 10 -](#_Toc26657544)

[4.3.1测试用例trace01.txt - 10 -](#_Toc26657545)

[4.3.2测试用例trace02.txt - 11 -](#_Toc26657546)

[4.3.3测试用例trace03.txt - 11 -](#_Toc26657547)

[4.3.4测试用例trace04.txt - 11 -](#_Toc26657548)

[4.3.5测试用例trace05.txt - 11 -](#_Toc26657549)

[4.3.6测试用例trace06.txt - 12 -](#_Toc26657550)

[4.3.7测试用例trace07.txt - 12 -](#_Toc26657551)

[4.3.8测试用例trace08.txt - 12 -](#_Toc26657552)

[4.3.9测试用例trace09.txt - 13 -](#_Toc26657553)

[4.3.10测试用例trace10.txt - 13 -](#_Toc26657554)

[4.3.11测试用例trace11.txt - 13 -](#_Toc26657555)

[4.3.12测试用例trace12.txt - 14 -](#_Toc26657556)

[4.3.13测试用例trace13.txt - 14 -](#_Toc26657557)

[4.3.14测试用例trace14.txt - 14 -](#_Toc26657558)

[4.3.15测试用例trace15.txt - 15 -](#_Toc26657559)

[4.4 自测试评分 - 15 -](#_Toc26657560)

[第5章 总结 - 16 -](#_Toc26657561)

[5.1 请总结本次实验的收获 - 16 -](#_Toc26657562)

[5.2 请给出对本次实验内容的建议 - 16 -](#_Toc26657563)

[参考文献 - 18 -](#_Toc26657564)

# 第1章 实验基本信息

## 1.1 实验目的

理解现代计算机系统进程与并发的基本知识

掌握 linux 异常控制流和信号机制的基本原理和相关 系统函数

掌握shell的基本原理和实现方法

深入理解Linux信号响应可能导致的并发冲突及解决方法

培养Linux下的软件系统开发与测试能力

## 1.2 实验环境与工具

### 1.2.1 硬件环境

X64CPU；1.6GHz；8G RAM；512G SSD

### 1.2.2 软件环境

Windows10 64位；Ubuntu 18.04.5 LTS 64位

### 1.2.3 开发工具

Clion2020.3；Vscode；gedit+gcc

## 1.3 实验预习

上实验课前，必须认真预习实验指导书（PPT或PDF）

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。

了解进程、作业、信号的基本概念和原理

了解shell的基本原理

熟知进程创建、回收的方法和相关系统函数

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）

进程：一个执行程序中的实例，进程给应用程序提供了两个关键的抽象：（1）逻辑控制流，每个应用程序似乎独占CPU，这由OS内核通过上下文切换机制完成。（2）私有地址空间，每个程序似乎独占整个内存空间，这由OS内核通过虚拟地址实现。

创建进程：使用fork函数，父进程调用fork函数创建一个新的运行的子进程，子进程得到和父进程用户及虚拟地址空间完全相同的一个副本。

回收方法：（1）当一个进程由于某种原因终止时，进程保持在一种已经终止的状态，直到被他的父进程回收，当父进程回收他的子进程时，内核将子进程的退出状态传递给父进程，然后抛弃已经终止的进程，从此开始，该子进程就不存在了。（2）如果一个父进程终止了，内核会安排init进程成为它的孤儿进程的父亲，init进程将会回收这些孤儿进程。

## 2.2信号的机制、种类（5分）

信号的机制：一个信号就是一条消息，它通知进程系统中发生了一个某种类型的事件。类似于异常和中断，一般是从内核发送到一个进程（有时是在另个一进程的请求下）。每种信号类型都对应某种系统的事件。信号类型是用小整数ID来标识，每一个信号对应唯一的ID。信号唯一的信息是它的ID和它的到达。

发送信号：内核通过更新目的进程上下文的某种状态，发送一个信号给目的进程。

接收信号：内核强迫目的进程以某种方式对信号的发送做出反应，则接收了这种信号。

信号的类型



## 2.3 信号的发送方法、阻塞方法、处理程序的设置方法（5分）

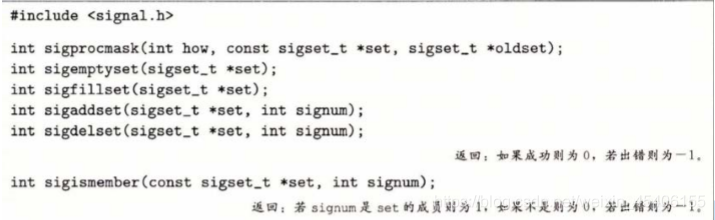
信号的发送方法：

1. 用/bin/kill程序发送信号，/bin/kill程序可以向另外的进程发送任意的信号
2. 从键盘发送信号，在键盘上输入Ctrl+C/Z会导致内核发送一个SIGINT/SIGSTOP信号到前台进程中的每个进程，默认情况下是终止前台作业
3. 用kill函数发送信号给其他进程（包括他们自己）
4. 使用alarm函数发送信号，进程可以通过调用 alarm 函数在指定 secs 秒后发送一个 SIGALRM 信号给调用进程。

信号的阻塞方法：

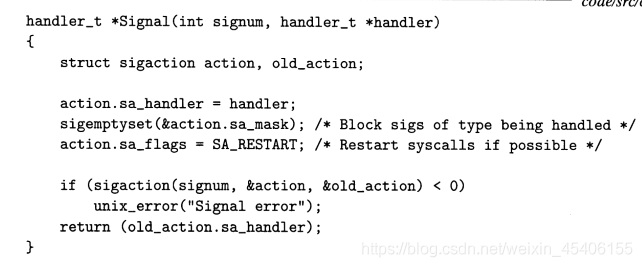
隐式阻塞机制：内核默认阻塞任何当前处理程序正在处理信号类和待处理信号。

显示阻塞进制：应用程序可以调用 sigprocmask 函数和它的辅助函数，明确地阻塞和解除阻塞选定的信号。调用要求如下：



处理程序的设置方法：

1. 调用 signal 函数，调用 signal(SIG,handler)，SIG 代表信号类型，handler 代表接收到 SIG 信号之后对应的处理程序。
2. 因为 signal 的语义各有不同，所以我们需要一个可移植的信号处理函 数设置方法，Posix 标准定义了 sigaction 函数，它允许用户在设置信号。如下：



## 2.4 什么是shell，功能和处理流程（5分）

shell：shell是一个用c语言编写的程序，它是用户使用Linux的桥梁，shell既是一种命令语言，又是一种程序设计语言，shell是一种应用程序。

功能：shell应用程序提供了一个界面，用户通过访问这个界面访问操作系统内 核的服务。

处理流程：

1. 从终端读入输入的命令。
2. 将输入字符串切分获得所有的参数。
3. 如果是内置命令则立即执行，否则调用相应的程序执行。
4. shell 应该接受键盘输入信号，并对这些信号进行相应处理。

# 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：解析命令行，如果不是内置命令，则fork子进程执行之。是则执行。

参 数：cmdline，命令行参数。

处理流程：首先调用parseline函数解析命令行，如果是空行则跳过。然后判断是否内置命令。不是的话首先阻塞命令行参数，防止其与添加job到jobs里产生竞争，然后fork一个子进程，子进程运行命令行中的程序。tsh则将子进程放入另外一个进程组中，然后解除阻塞。

要点分析：

1. 在创建子进程并运行前，首先需要将SIGCHLD，SIGINT，SIGSTP信号阻塞，因为这些信号处理函数会调用job相关函数，这些函数会对jobs进行操作，而接下来父进程还要fork出子进程，将之加入到jobs，所以为了避免子进程与信号处理函数竞争jobs，需要阻塞上述信号。处理完之后需要解除阻塞。同时，子进程从父进程继承了信号阻塞向量，所以子进程必须确保在执行新程序之前接触对信号的阻塞。
2. tsh是在Linux 的shell（bash）下运行的，tsh在前台进程组中运行，此时，tsh创建的子进程也默认在前台进程组中。因此ctrl-c会给shell(bash)下前台进程组中的所有进程发送SIGINT信号，包括tsh和tsh创建的进程，这样显然不好。因此在fork出子进程后、execve程序前，子进程调用函数setpgid(0, 0)将自己放到一个新的进程组中（进程组ID与子进程的PID相同)。在前台进程组中只有一个进程tsh。当键入Ctrl-C， shell(bash)将捕获产生的SIGINT，然后转发给tsh， tsh收到SIGINT后，转发给适当的前台作业（更准确的说法：前台作业的进程组）

## 3.1.2 int builtin\_cmd(char \*\*argv)函数（5分）

函数功能：判断命令是否是内置指令，是的话立即执行，不是的话返回，对单独的“&”无视。

参 数：char\*\* argv

处理流程：判断输入的参数是否是“quit”“bg”“fg”“jobs”，是的话调用对应的函数并返回1，不是的话返回0。

要点分析：quit的话直接调用exit(0)，不返回1。

## 3.1.3 void do\_bgfg(char \*\*argv) 函数（5分）

函数功能：实现内置命令bg，fg。

参 数：char\*\* argv。

处理流程：

1. 首先对输入的参数进行解析，处理没有指定进程/job的参数，找不到进程/job的情况，然后判断命令是bg还是fg。
2. 如果是bg，发送SIGCONT信号，state设为BG，打印信息。如果是fg则发送信号SIGCONT，状态设为FG，打印信息，调用waitfg函数处理指定进程。

要点分析：

1. 注意处理error，exception，提高程序的健壮性。
2. 不仅要发送信号，也不能忘记设置其状态为BG/FG。
3. 记得fg命令要调用waitfg阻塞tsh运行，使得前台一直都是这个job直到结束。

## 3.1.4 void waitfg(pid\_t pid) 函数（5分）

函数功能：阻塞直到指定 pid 的进程不再是前台进程。

参 数：pid\_t pid

处理流程：进行 while 循环，每次 循环 sleep 1 秒，直到前台程序的 PID 不再是 该pid。

要点分析：无。

## 3.1.5 void sigchld\_handler(int sig) 函数（10分）

函数功能：父进程中接收到 SIGCHLD 信号的处理函数。

参 数：int sig

处理流程：

1. 记录下旧error
2. 处理所有子进程集合中已经停止或者终止的子进程。
3. 如果该子进程通过调用 exit 或者一个返回正常终止，则阻塞信号，删 除 job，恢复信号。
4. 如果该子进程当前已经停止，向屏幕打印信息。
5. 如果该当前子进程是因为一个未被捕获的信号终止的，则向屏幕打印 信息，阻塞信号，删除 job，恢复信号。
6. 恢复errno。

要点分析：

1. 注意保存恢复error

**3.2 程序实现（tsh.c的全部内容）（10分）**

见附件

**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**

# 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)，并填写完成4.3节的相应表格。

## 4.2 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

填写以下各个测试用例的测试结果，每个测试用例1分。

### 4.3.1测试用例trace01.txt

|  |  |  |
| --- | --- | --- |
| tsh测试结果 | | tshref测试结果 |
| 2021-06-10 22-09-50 的屏幕截图 | | 2021-06-10 22-20-43 的屏幕截图 |
| 测试结论 | 相同 | |

### 4.3.2测试用例trace02.txt

|  |  |  |
| --- | --- | --- |
| tsh测试结果 | | tshref测试结果 |
| 2021-06-10 22-10-09 的屏幕截图 | | 2021-06-10 22-20-51 的屏幕截图 |
| 测试结论 | 相同 | |

### 4.3.3测试用例trace03.txt

|  |  |  |
| --- | --- | --- |
| tsh测试结果 | | tshref测试结果 |
| 2021-06-10 22-10-43 的屏幕截图 | | 2021-06-10 22-21-01 的屏幕截图 |
| 测试结论 | 相同 | |

### 4.3.4测试用例trace04.txt

|  |  |  |
| --- | --- | --- |
| tsh测试结果 | | tshref测试结果 |
| 2021-06-10 22-10-58 的屏幕截图 | | 2021-06-10 22-21-16 的屏幕截图 |
| 测试结论 | 相同 | |

### 4.3.5测试用例trace05.txt

|  |  |  |
| --- | --- | --- |
| tsh测试结果 | | tshref测试结果 |
| 2021-06-10 22-11-08 的屏幕截图 | | 2021-06-10 22-21-26 的屏幕截图 |
| 测试结论 | 相同 | |

### 4.3.6测试用例trace06.txt

|  |  |  |
| --- | --- | --- |
| tsh测试结果 | | tshref测试结果 |
| 2021-06-10 22-11-19 的屏幕截图 | | 2021-06-10 22-21-43 的屏幕截图 |
| 测试结论 | 相同 | |

### 4.3.7测试用例trace07.txt

|  |  |  |
| --- | --- | --- |
| tsh测试结果 | | tshref测试结果 |
| 2021-06-10 22-11-29 的屏幕截图 | | 2021-06-10 22-21-51 的屏幕截图 |
| 测试结论 | 相同 | |

### 4.3.8测试用例trace08.txt

|  |  |  |
| --- | --- | --- |
| tsh测试结果 | | tshref测试结果 |
| 2021-06-10 22-12-36 的屏幕截图 | | 2021-06-10 22-22-04 的屏幕截图 |
| 测试结论 | 相同 | |

### 4.3.9测试用例trace09.txt

|  |  |  |
| --- | --- | --- |
| tsh测试结果 | | tshref测试结果 |
| 2021-06-10 22-12-47 的屏幕截图 | | 2021-06-10 22-22-32 的屏幕截图 |
| 测试结论 | 相同 | |

### 4.3.10测试用例trace10.txt

|  |  |  |
| --- | --- | --- |
| tsh测试结果 | | tshref测试结果 |
| 2021-06-10 22-12-56 的屏幕截图 | | 2021-06-10 22-22-41 的屏幕截图 |
| 测试结论 | 相同 | |

### 4.3.11测试用例trace11.txt

测试中ps指令的输出内容较多，仅记录和本实验密切相关的tsh、mysplit等进程的部分信息即可。

|  |  |  |
| --- | --- | --- |
| tsh测试结果 | | tshref测试结果 |
| 2021-06-10 22-18-19 的屏幕截图 | | 2021-06-10 22-22-50 的屏幕截图 |
| 测试结论 | 相同 | |

### 4.3.12测试用例trace12.txt

测试中ps指令的输出内容较多，仅记录和本实验密切相关的tsh、mysplit等进程的部分信息即可。

|  |  |  |
| --- | --- | --- |
| tsh测试结果 | | tshref测试结果 |
| 2021-06-10 22-18-44 的屏幕截图 | | 2021-06-10 22-23-08 的屏幕截图 |
| 测试结论 | 相同 | |

### 4.3.13测试用例trace13.txt

测试中ps指令的输出内容较多，仅记录和本实验密切相关的tsh、mysplit等进程的部分信息即可。

|  |  |  |
| --- | --- | --- |
| tsh测试结果 | | tshref测试结果 |
| 2021-06-10 22-19-05 的屏幕截图 | | 2021-06-10 22-23-47 的屏幕截图 |
| 测试结论 | 相同 | |

### 4.3.14测试用例trace14.txt

|  |  |  |
| --- | --- | --- |
| tsh测试结果 | | tshref测试结果 |
| 2021-06-10 22-19-50 的屏幕截图 | | 2021-06-10 22-24-38 的屏幕截图 |
| 测试结论 | 相同 | |

### 4.3.15测试用例trace15.txt

|  |  |  |
| --- | --- | --- |
| tsh测试结果 | | tshref测试结果 |
| 2021-06-10 22-20-04 的屏幕截图 | | 2021-06-10 22-24-56 的屏幕截图 |
| 测试结论 | 相同 | |

# 第5章 评测得分

**总分20分**

实验程序统一测试的评分（教师评价）：

（1）正确性得分：

（2）性能加权得分：

# 第6章 总结

## 5.1 请总结本次实验的收获

更深了了解计算机进程，exceptions，信号等。

## 5.2 请给出对本次实验内容的建议

有点难，还好给出了几个函数。

注：本章为酌情加分项。

# 参考文献

**为完成本次实验你翻阅的书籍与网站等**

[1] 林来兴. 空间控制技术[M]. 北京：中国宇航出版社，1992：25-42.

[2] 辛希孟. 信息技术与信息服务国际研讨会论文集：A集[C]. 北京：中国科学出版社，1999.

[3] 赵耀东. 新时代的工业工程师[M/OL]. 台北：天下文化出版社，1998 [1998-09-26]. http://www.ie.nthu.edu.tw/info/ie.newie.htm（Big5）.

[4] 谌颖. 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.