

ICS-LAB7 TinyShell

微壳

哈尔滨工业大学
计算机科学与技术学院

2021年6月2日, Wednesday

一、实验基本信息

■ 实验类型：设计型实验

■ 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

■ 实验分组

- 一人一组

一、实验基本信息

- 实验学时： 3
- 实验分数： 5，本次实验按100分计算，折合成总成绩的5分。
- 实验地点： G712、 G709
- 实验环境与工具：
 - X64 CPU； 2GHz； 2G RAM； 256GHD Disk 以上
 - Windows7 64位以上； VirtualBox/Vmware 11以上； Ubuntu 16.04 LTS 64位/优麒麟 64位

一、实验基本信息

- 学生实验准备：禁止准备不合格的学生做实验
 - 个人笔记本电脑
 - 实验环境与工具所列明软件
 - 参考手册：Linux环境下的命令；GCC手册；GDB手册
 - <http://docs.huihoo.com/c/linux-c-programming/> C汇编Linux手册
 - <http://csapp.cs.cmu.edu/3e/labs.html> CMU的实验参考
 - <http://www.linuxidc.com/> <http://cn.ubuntu.com/>
<http://forum.ubuntu.org.cn/>

二、实验要求

- 学生应穿鞋套进入实验室
- 进入实验室后在签到簿中签字
- 实验安全与注意事项
 - 禁止使用笔记本电脑以外的设备
 - 学行生不得自行开关空调、投影仪
 - 学生不得自打开窗户
 - 不得使用实验室内的其他实验箱、示波器、导线、工具、遥控器等
 - 认真阅读消防安全撤离路线
 - 突发事件处理：第一时间告知教师，同时关闭电源插排开关。
- 遵守学生实验守则，爱护实验设备，遵守操作规程，精心操作，注意安全，严禁乱拆乱动。
- 实验结束后要及时关掉电源，对所用实验设备进行整理，设备摆放和状态恢复到原始状态。
- 桌面整洁、椅子归位，经实验指导教师允许后方可离开

三、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

三、实验预习

■ Kill命令

- `kill -l`: 列出信号
- `kill -SIGKILL 17130`: 杀死pid为17130的进程
- `kill -9 17130`: 杀死pid为17130的进程, 或者:
- `kill -9 -17130`: 杀死进程组17130中的每个进程
- `killall -9 pname`: 杀死名字为pname的进程

三、实验预习

■ 进程状态

- D 不可中断睡眠 (通常是在IO操作) 收到信号不唤醒和不可运行, 进程必须等待直到有中断发生
- R 正在运行或可运行 (在运行队列排队中)
- S 可中断睡眠 (休眠中, 受阻, 在等待某个条件的形成或接受到信号)
- T 已停止的 进程收到SIGSTOP, SIGTSTP, SIGTTIN, SIGTTOU信号后停止运行
- W 正在换页(2.6.内核之前有效)
- X 死进程 (未开启)
- Z 僵尸进程a defunct ("zombie") process
- < 高优先级(not nice to other users)
- N 低优先级(nice to other users)
- L 页面锁定在内存 (实时和定制的IO)
- s 一个信息头
- l 多线程 (使用 CLONE_THREAD, 像NPTL的pthreads的那样)
- + 在前台进程组

三、实验预习

■ **ps t /ps aux /ps**

- **t** <终端机编号n> 列终端n的程序的状况。
- **a** 显示现行终端机下的所有程序，包括其他用户的程序。
- **u** 以用户为主的格式来显示程序状况。
- **x** 显示所有程序，不以终端来区分。

```
Linux>sleep 2000 |more|sort|grep hit &
```

```
Linux>ps -f a
```

```
Linux>ps aj
```

三、实验预习

- 作业：`jobs`、`fg %n`、`bg %n`
 - `jobs` 显示当前暂停的进程
 - `bg %n` 使第n个任务在后台运行(%前有空格)
 - `fg %n` 使第n个任务在前台运行
 - `bg, fg` 不带`%n` 表示对最后一个进程操作
 - `ctrl+c`: 终止前台作业(进程组的每个进程)
 - `ctrl+z`: 停止前台作业(进程组的每个进程), 随后可用`bg`恢复后台运行, `fg` 恢复前台运行。

四、实验内容与步骤

■ 1.环境建立

- Ubuntu + gcc

■ 2.获得实验包

- 从实验教师处获得下 shlab-handout-hit.tar
- 也可以从课程QQ群下载，也可以从其他同学处获取。
- **HIT与CMU的不同**

■ 3. 实验报告解压（linux下）

解压命令 `unix>tar xvf shlab-handout-hit.tar`

4. 实验包内容介绍

■ 数据包中包含下面文件：

- tsh.c : tiny-shell 的代码框架，要求实现里面的空函数
- tshref: 参考答案的程序（可执行文件），用于对比程序行为，验证实验代码tsh.c的正确性：

tiny-shell的输出应该与tshref的输出完全一致

- 16个轨迹文件(trace file): trace01.txt trace16.txt
- sdriver.pl: shell驱动程序，以子进程的方式运行shell，并根据轨迹文件向shell发送命令和信号。
 - 获得帮助: `unix> ./sdriver.pl -h`

5.实验任务

■ 完成tsh.c中的空函数

tsh.c : tiny-shell 的代码框架, 包含了基本的代码。

- tsh.c的头部注释段写入名字
- **空函数是实验要求实现的部分**
- ✓ **eval: 解析和解释命令行的主例程。 [70行] (已提供答案)**
- ✓ builtin_cmd: 识别并解释内置命令: **quit, fg, bg, 和 jobs**. [25行]
- ✓ **do_bgfg: 实现内置命令bg 和 fg. [50 行] (已提供答案)**
- ✓ waitfg: 等待一个前台作业结束. [20 行]
- ✓ sigchld_handler: 捕获SIGCHILD信号. [80 行]
- ✓ sigint_handler: 捕获SIGINT (ctrl-c) 信号. [15 行]
- ✓ sigtstp_handler: 捕获SIGTSTP (ctrl-z) 信号. [15 行]
- 编译链接: `unix> make`
- 运 行: `unix> ./tsh`

6 任务要求

- tsh的提示符： `tsh>`
- 用户输入的命令行应该包括一个名字、0或多个参数，并用一个或多个空格分隔。
- 如果名字是内置命令，tsh立即处理并等待用户输入下一个命令行。
- 否则，假定这个名字是一个可执行文件的路径，tsh在初始子进程的上下文中加载和运行它。
- tsh不需要支持管（|）或I/O重定向（<和>）。是指这个初始子进程

6 任务要求

- 键入ctrl-c/ctrl-z，应该把信号SIGINT/SIGTSTP发送到当前的前台作业，及该作业的子孙作业（例如，它创建的任何子进程）。如果没有前台工作，那么信号应该没有效果。
- 如果命令行以&结尾，则tsh在后台运行该作业；否则，在前台运行该作业
- 可以用进程ID(PID)或tsh赋予的正整数作业ID(job ID, JID)标识一个作业。JID用前缀%，例如“%5”标识作业ID为5的作业，“5”表示JID为5的作业。
- 已经提供了处理作业列表所需的所有函数

#

[trace10.txt](#) - Process fg builtin command.

#

/bin/echo -e tsh> ./myspin 4 \046

./myspin 4 &

SLEEP 1

/bin/echo tsh> fg %1

fg %1

SLEEP 1

TSTP

/bin/echo tsh> jobs

jobs

/bin/echo tsh> fg %1

fg %1

/bin/echo tsh> jobs

jobs

有用的测试小程序：

■ myint <n>

n次调用sleep(1)，睡眠n秒后，用kill函数给自己发送信号SIGINT后退出。

■ myspin <n>

n次调用sleep(1)，睡眠n秒后，退出。

■ mysplit <n>

fork子进程后，等待回收子进程；子进程n次调用sleep(1)，然后退出。

■ mystop <n>

n次调用sleep(1)，睡眠n秒后，给自己发送信号SIGTSTP, ...。

7 程序测试

- 用shell驱动程序sdriver.pl和追踪文件(trace file)测试
- 测试你的shell程序tsh

```
unix> ./sdriver.pl -t trace01.txt -s ./tsh -a "-p"
```

参数-a "-p" 告诉shell不发送命令提示符

或：

```
unix> make test01
```

- 测试参考shell程序tshref

```
unix> ./sdriver.pl -t trace01.txt -s ./tshref -a "-p"
```

或：

```
unix> make rtest01
```

7 程序测试

- tshref.out已经给出了参考shell程序在所有trace file上的输出，方便查阅。
- 使用trace file进行测试，除了有文件头注释段落信息外，其余均与手工交互测试的输出结果相同。
- 建议：从trace01.txt开始验证，没有问题后，在依次验证trace02.txt, trace03.txt ...

8 建议

■ 需要熟知的函数

- waitpid函数及其选项 WUNTRACED 、 WNOHANG
- kill
- fork
- execve
- setpgid
- waitpid、 wait
- signal
- sigprocmask
- ...

8 建议

■ 需要熟知的函数...

`int kill(pid_t pid, int signo);`

a、`pid > 0` ， 信号发送给pid进程；

b、`pid == 0`， 把信号发送给本进程（自己）所在的进程组中所有进程，不包括系统进程；

c、`pid < 0` ， 把信号发送给进程组-pid中的所有进程；

d、`pid == -1`， 把信号发送给除系统进程外的所有进程（有些进程不接受9和19号信号）。

8 建议

- waitfg函数和SIGCHLD信号处理程序的分工配合需要斟酌
- 建议：
 - 在waitfg函数中，在sleep函数附近使用busy loop，例如：
while (xxxxx) sleep(1);
 - 仅在SIGCHLD处理程序中回收进程，程序逻辑清晰、简单
 - 在SIGCHLD处理程序中，采用非挂起的方式(非死等)调用waitpid函数，如：

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

```
{/*立即返回，返回“停止”或“终止”的进程id */.....}
```

8 建议

- 防止竞争: 父进程调用addjob之前, 防止子进程被信号处理程序回收的方法:
 - 在eval中, 父进程必须在用fork创建子进程前, 使用sigprocmask阻塞SIGCHLD信号
 - 父进程创建完成子进程并用addjob记录后, 用sigprocmask解除阻塞。
 - 子进程从父进程处继承了信号阻塞向量, 子进程必须确保在执行新程序之前解除对SIGCHLD的阻塞。
- 不要在tsh中运行more, less, vi, emacs等程序 (这些程序利用终端terminal设置做一些比较奇特的事情)
- 运行基于简单文本的程序: /bin/ls, /bin/ps, /bin/echo

8 建议

- 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。
- 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——这样显然不对。
- 解决办法：
 - 在fork后、execve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）

五、实验报告格式与评分

■ 实验报告格式

按照实验报告模板所要求的格式与内容书写。

■ 评 分

本次实验成绩按100分计

- 按时上课，签到5分
- 按时下课，不早退5分
- 课堂表现：10分，不按操作规程、非法活动扣分。
- 实验报告：80分。具体参见实验报告各环节的分值
- 在实验报告中，对每一任务，按照要求用文字详细描述
- 杜绝抄袭！发现全0分！

9.实验提交

- 提交内容——3个文件：
 - tsh.c文件（非压缩格式）
 - *教师将使用自动评分工具，对代码进行自动评测（满分80），将测试结论评分按测试评分/80*30的方式折算。*
 - 实验报告文件word版（填写4.4 自测试评分）
 - 实验报告pdf版
- **提交时间：实验后 1周内提交，迟交扣分！**
- 提交方式1：乐学网按提示提交
- 提交方式2：
 - 学生提交1个压缩包即可
 - 课代表提交1个包给授课教师