

《操作系统》实验

一、Shell及系统调用

助教：彭小双、郑学森

指导教师：翁楚良

2023 春 ECNU

目标

1. 安装MINIX操作系统(Version:3.3)
2. 学习Shell, 系统编程, 实现一个基本的Shell

实验要求

■ Shell能解析的命令如下：

1. 带参数的程序运行功能。

`program arg1 arg2 ... argN`

2. 重定向功能，将文件作为程序的输入/输出。

1. ">" 表示覆盖写

`program arg1 arg2 ... argN > output-file`

2. ">>" 表示追加写

`program arg1 arg2 ... argN >> output-file`

3. "<" 表示文件输入

`program arg1 arg2 ... argN < input-file`

实验要求(续)

■ Shell能解析的命令行如下：

3. 管道符号 “|” ， 在程序间传递数据。

`programA arg1 ... argN | programB arg1 ... argN`

4. 后台符号& ,表示此命令将以后台运行的方式执行。

`program arg1 arg2 ... argN &`

5. 工作路径移动命令cd。

6. 程序运行统计mytop。

7. shell退出命令exit。

8. history n显示最近执行的n条指令。

实现说明

■ Shell主体：

Shell主体结构是一个while循环，不断地接受用户键盘输入行并给出反馈。Shell将输入行分解成单词序列，根据命令名称分为二类分别处理，即**shell内置命令**（例如cd, history, exit）和**program命令**（例如/bin/目录下的ls, grep 等）。识别为shell内置命令后，执行对应操作。接受program命令后，利用minix自带的程序创建一个或多个新进程，并等待进程结束。如果末尾包含&参数，Shell可以不等待进程结束，直接返回。

实现说明(续)

■ Shell内置命令：

1. cd：因为Shell也是一个程序，启动时minix会分配一个当前工作目录，利用**chdir**系统调用可以移动Shell的工作目录。
2. history：保存Shell每次的输入行，打印所需字符串即可。
3. exit：退出Shell的while循环，结束Shell的main函数。

实现说明(续)

■ Shell内置命令：

4. mytop：参考minix终端输入top命令的输出信息，在minix系统/proc文件夹中通过open/read系统调用输出进程信息。
 1. /proc/meminfo中，查看内存信息，每个参数对应含义依次是页面大小pagesize，总页数量total，空闲页数量free，最大页数量largest，缓存页数量cached。可计算内存大小： $(pagesize * total) / 1024$ ，同理算出其他页内存大小。
 2. /proc/kinfo中，查看进程和任务数量。
 3. /proc/pid/psinfo中，例如 /proc/107/psinfo文件中，查看pid为107的进程信息。每个参数对应含义依次是：版本version，类型type，端点endpt，名字name，状态state，阻塞状态blocked，动态优先级priority，滴答ticks，高周期highcycle，低周期lowcycle，内存memory，有效用户ID effuid，静态优先级nice等。其中会用到的参数有：类型，状态，滴答。进程时间 $time = ticks / (u32_t)60$ 。
 4. 输出内容：
 1. 总体内存大小，空闲内存大小，缓存大小。
 2. 总体CPU使用占比。计算方法：得到进程和任务总数量total_proc，对每一个proc的ticks累加得到总体ticks，再计算空闲的ticks，最终可得到CPU使用百分比。

实现说明(续)

■ Program命令:

1. 运行程序: 利用fork调用创建进行子进程, 利用execvp调用将minix程序装载到该进程, 并赋予运行参数, 最后Shell利用wait/waitpid调用等待子进程结束。(参见UNIX高级编程8.3, 8.7和8.10节)
2. 重定向: minix为每个进程赋予键盘输入和控制台输出的文件描述符默认为0和1。子进程装载程序前, 利用close(0 or 1)将默认输入或者输出关闭, 再调用dup(fd)将某个打开文件的文件描述fd映射到标准输入或输出。(参见UNIX高级编程3.12节)
3. 管道: 若有n个子进程组成管道流, Shell在fork前先用pipe调用创建n-1对管道描述符, 关闭不需要的读写端。Shell运行fork后, 每个子进程利用dup将前一个管道的读端映射到标准输入, 后一个管道的写端映射到标准输出。(参见UNIX高级编程15.2节)
4. 后台运行: 为了屏蔽键盘和控制台, 子进程的标准输入、输出映射成/dev/null。子进程调用signal(SIGCHLD, SIG_IGN), 使得minix接管此进程。因此Shell可以避免调用wait/waitpid直接运行下一条命令。

参考资料

1. 在实现管道时，需要用到pipe和dup系统调用等，重定向标准向输入输出(参见教材1.4.3节)。
2. Shell主要是为用户提供一个命令解释器，接收用户命令（如ls等），然后调用相应的应用程序。实现的Shell支持后台进程的运行。
3. 在计算内存和CPU总体使用情况时，可参考top命令实现
<https://github.com/0xffea/MINIX3/blob/master/usr.bin/top/top.c>
4. 推荐《UNIX环境高级编程》查找详细的系统调用使用方法。

测试用例

1. `cd /your/path`
2. `ls -a -l`
3. `ls -a -l > result.txt`
4. `vi result.txt`
5. `grep a < result.txt`
6. `ls -a -l | grep a`
7. `vi result.txt &`
8. `mytop`
9. `history n`
10. `exit`

实验提交

1. 提交材料：

1. 安装、测试、和编译等过程说明文档；
2. Shell实现源码及Shell说明文档。

2. 实验要求：

1. 独立完成实验，严禁抄袭。
2. 请在规定时间内提交实验。