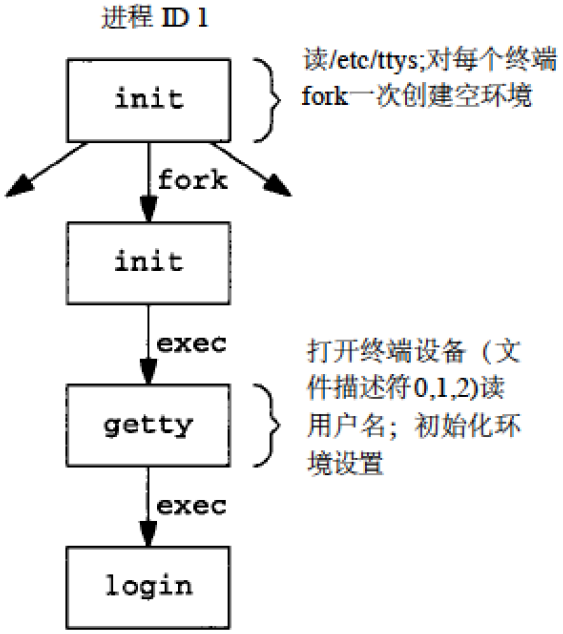
第九章 进程关系

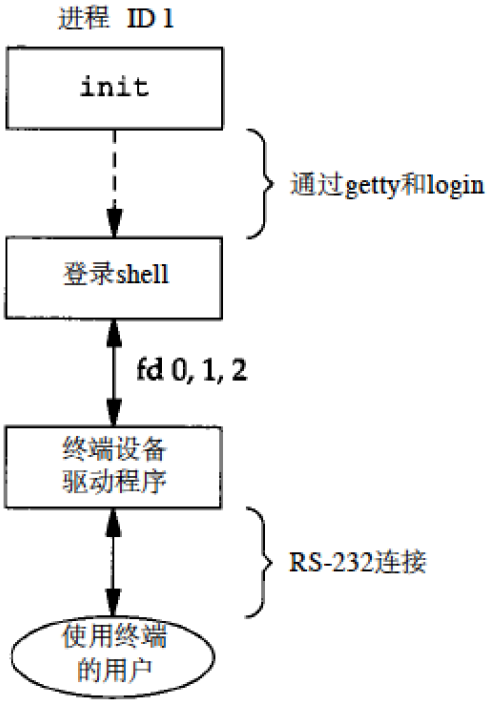
冯 晶 E-mail: gfeng@whu.edu.cn

1. 终端登录

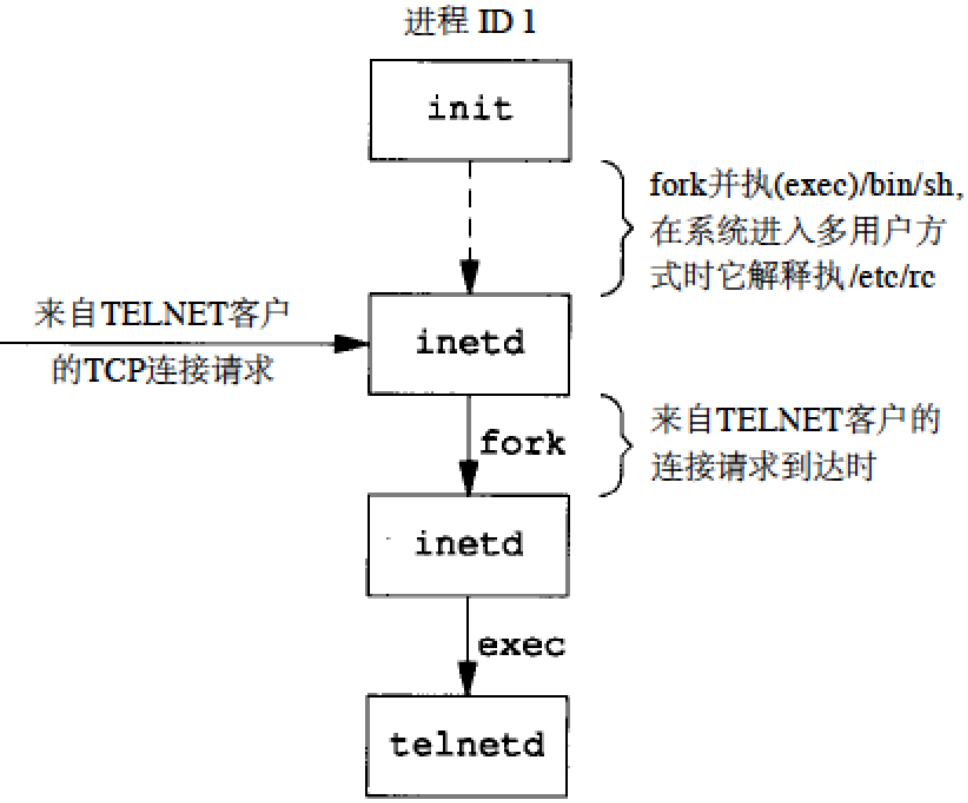
execle("/usr/bin/login", "login", "-p", username, (char *) 0, envp);

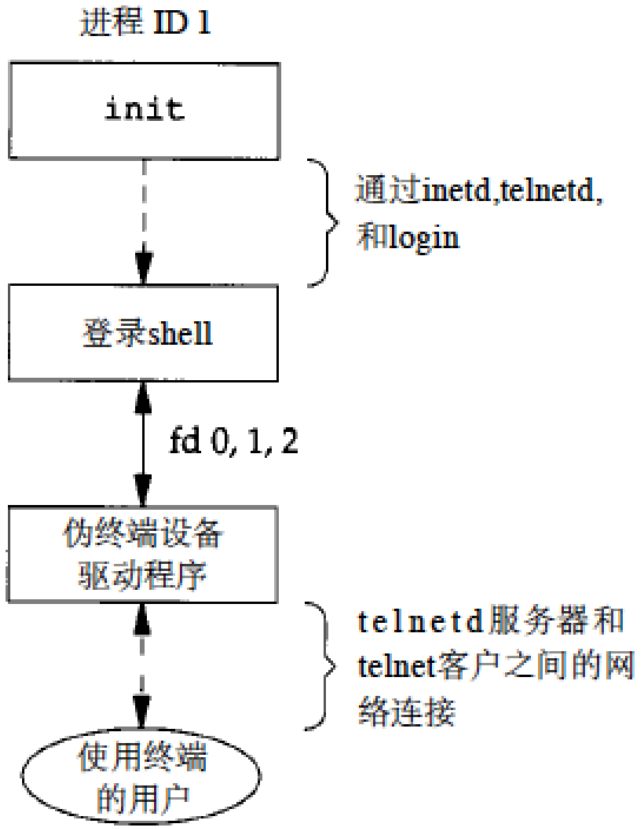
execl("/bin/sh", "-sh", (char *) 0);





2. 网络登录


进程 ID 1
init
fork并执(exec)/bin/sh,
在系统进入多用户方
式时它解释执/etc/rc
来自TELNET客户
的TCP连接请求
inetd
fork
来自TELNET客户的
连接请求到达时
inetd
exec
telnetd

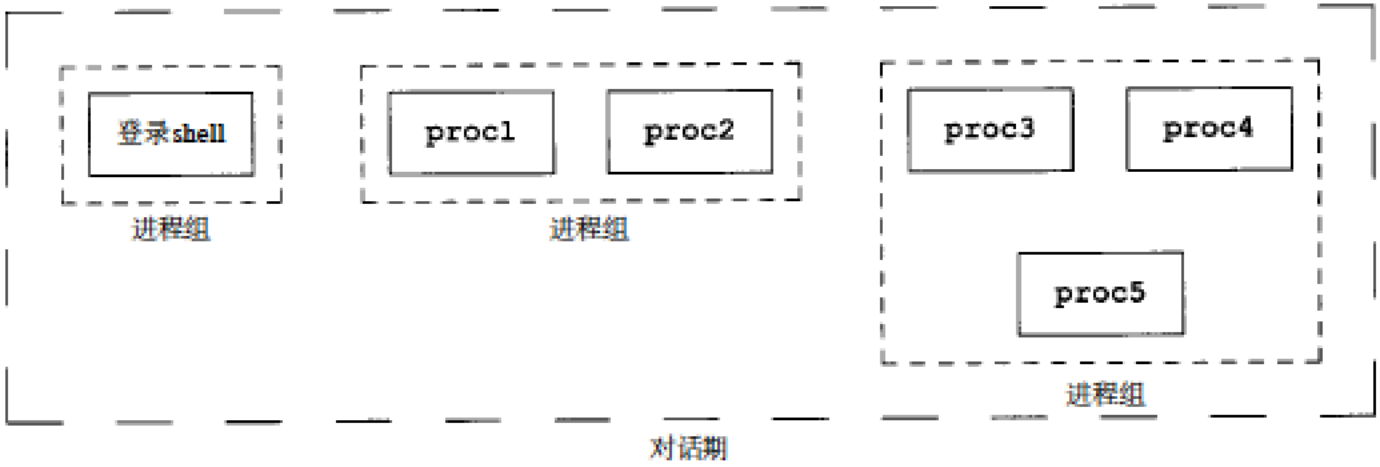

进程 ID 1
init
通过inetd,telnetd,
和login
登录shell
fd 0, 1, 2
伪终端设备
驱动程序
telnetd服务器和
telnet客户之间的网
络连接
使用终端
的用户

3. 进程组


```
#include <sys/types.h>
#include <unistd.h>

pid_t getpgrp(void);

int setpgid(pid_t pid, pid_t pgid);
```

4. 对话期


```
proc1 | proc2 &
proc3 | proc4 | proc5
```



4. 对话期

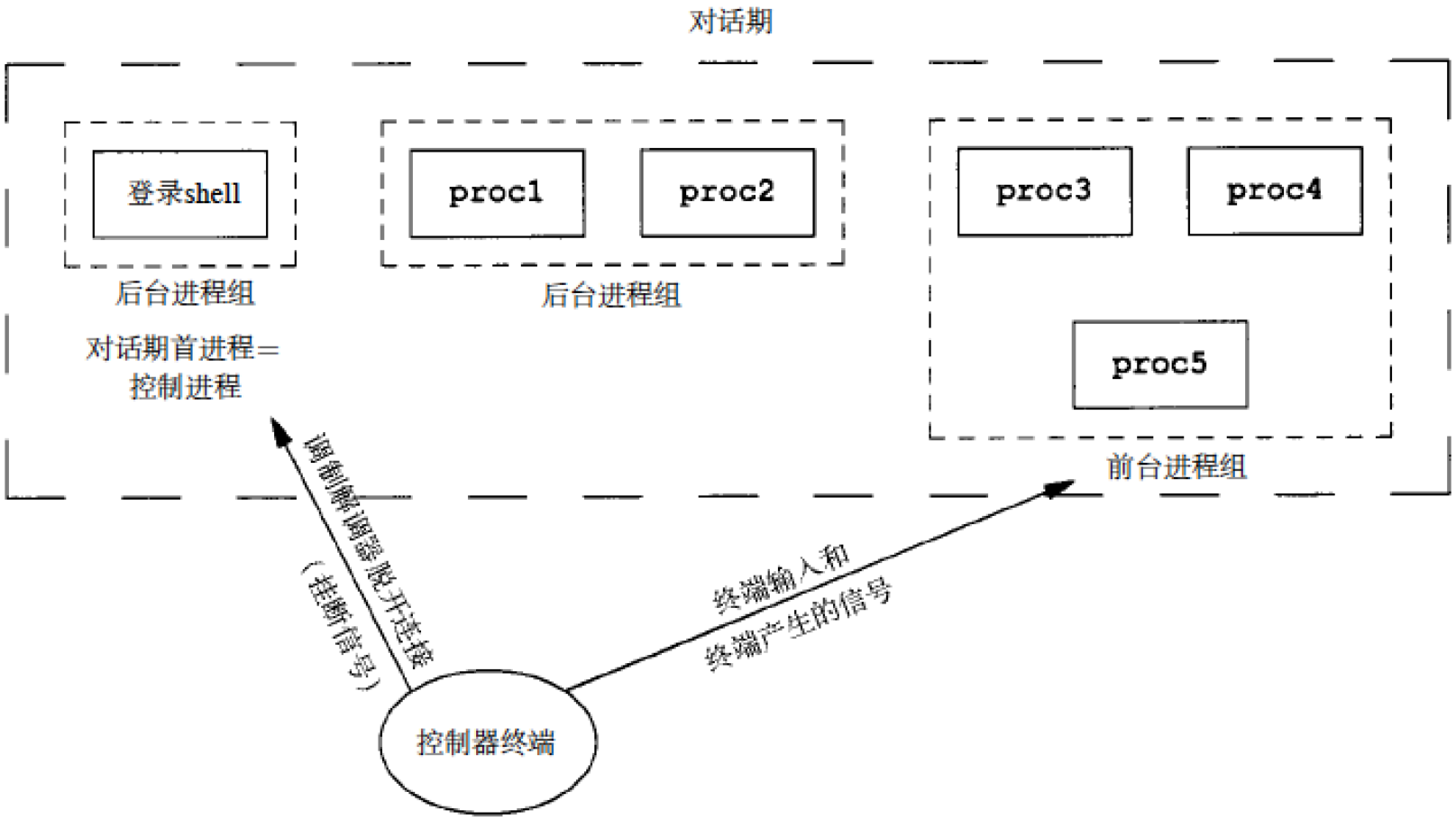

```
#include <sys/types.h>
#include <unistd.h>

pid_t  setsid(void);
```

- 创建新对话期
  - 此进程变为新对话期的对话期首进程，是该新对话期中的唯一进程
  - 此进程成为一个新进程组的组长进程
  - 此进程没有控制终端

5. 控制终端

- 一个对话期可以有一个独立的控制终端
- 建立与控制终端连接的对话期首进程，称为控制进程
- 一个对话期中的几个进程组可被分成一个前台进程组和若干个后台进程组
- 如果一个对话期有一个控制终端，则它有一个前台进程组，其它进程组则为后台进程组
- 键入中断健（Ctrl-C）或退出键（Ctrl-\），将中断信号或退出信号发送至前台进程组的所有进程
- 终端界面检测到网络已经断开，则将挂断信号发送至控制进程

5. 控制终端


对话期
登录shell
proc1
proc2
proc3
proc4
proc5
后台进程组
后台进程组
对话期首进程=
控制进程
前台进程组
调制解调器脱开连接
（挂断信号）
终端输入和
终端产生的信号
控制器终端

6. 作业控制

- 支持作业控制的shell
- 内核中的终端驱动程序必须支持作业控制
- 必须提供对某些作业控制信号的支持

```
$ make all > Make.out &
[1]      1475
$ pr *.c | lpr &
[2]      1490
$                                  键入回车
[2] + Done              pr *.c | lpr &
[1] + Done              make all > Make.out &
```

6. 作业控制

- 终端驱动程序产生信号
  - 中断字符（Ctrl-C）产生SIGINT
  - 退出字符（Ctrl-\）产生SIGQUIT
  - 挂起字符（Ctrl-Z）产生SIGTSTP

6. 作业控制

- 将作业转为前台作业运行

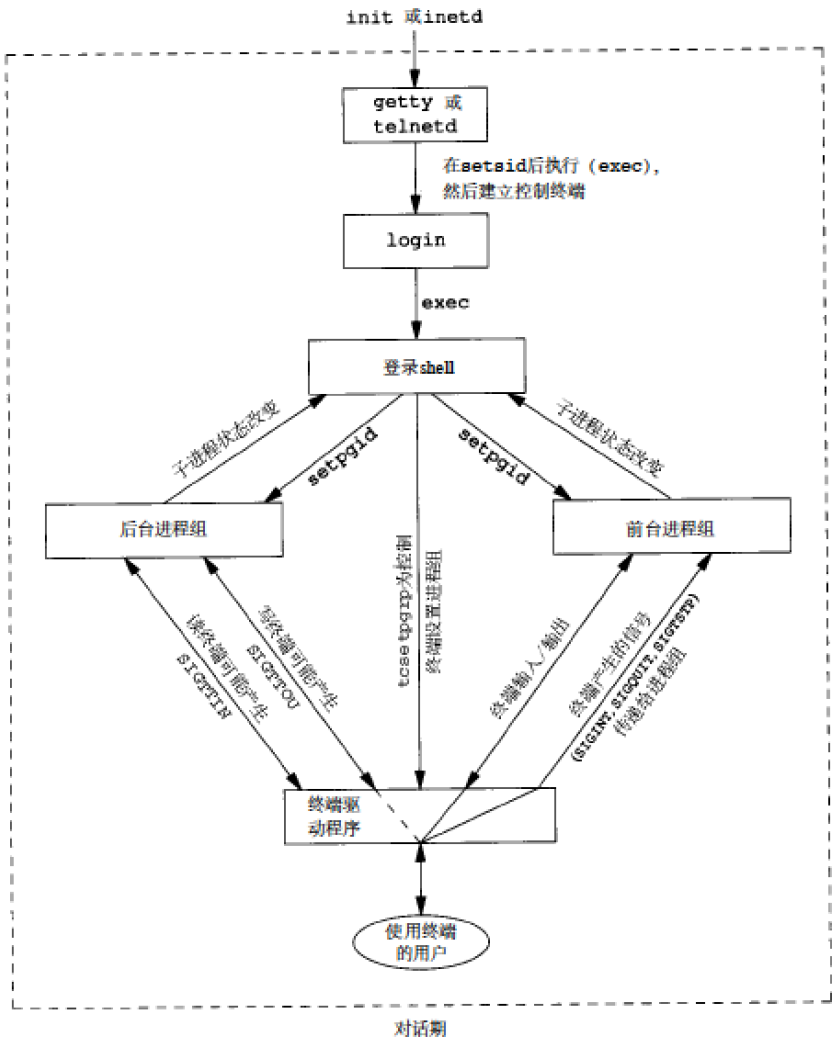
```
$ cat > temp.foo &          在后台启动，但将从标准输入读
[1]     1681
$                           键入回车
[1] + Stopped (tty input)       cat > temp.foo &
$ fg %1                     使1号作业成为前台作业
cat > temp.foo              shell告诉我们现在哪一个作业在前台
hello, world                输入1行
^D                          键入文件结束符
$ cat temp.foo              检查该行已送入文件
hello, world
```

6. 作业控制

- 后台作业输出到控制终端

```
$ cat temp.foo &          在后台执行
[1]      1719
$ hello, world            在提示符后出现后台作业的输出
                          键入回车
[1] + Done          cat temp.foo &
$ stty tostop             禁止后台作业向控制终端输出
$ cat temp.foo &          在后台再次执行
[1]      1721
$                         键入回车，发现作业已停止
[1] + Stopped(tty output)          cat temp.foo &
$ fg %1                   将停止的作业恢复为前台作业
cat temp.foo              shell告诉我们现在哪一个作业在前台
hello, world              该作业的输出
```

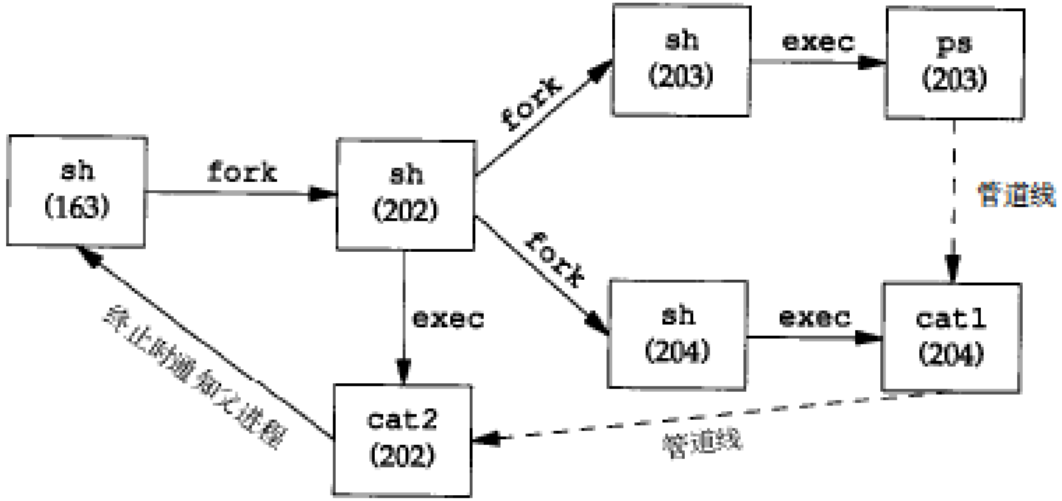
7. shell执行程序


init 或inetd
getty 或 telnetd
在setsid后执行（exec），然后建立控制终端
login
exec
登录shell
子进程状态改变
setpgid
setpgid
子进程状态改变
后台进程组
前台进程组
tcsetpgrp为控制终端设置进程组
读终端可能产生SIGTTIN
写终端可能产生SIGTTOU
终端输入/输出
终端产生的信号(SIGINT, SIGQUIT, SIGTSTP)传递给进程组
终端驱动程序
使用终端的用户
对话期

7. shell执行程序


```
ps -xj | cat1 | cat2
```

```
PPID   PID   PGID   SID   TPGID   COMMAND
   1   163    163   163     163   -sh
 163   202    163   163     163   cat2
 202   203    163   163     163   ps
 202   204    163   163     163   cat1
```



8. 孤儿进程组

- 一个父进程已终止的进程成为孤儿进程，这种进程由init进程收养。