Advanced Programming in the UNIX Environment

Week 07, Segment 2:
Process Groups and Sessions

Department of Computer Science
Stevens Institute of Technology

Jan Schaumann
jschauma@stevens.edu
https://stevens.netmeister.org/631/

Process Groups

```
#include <unistd.h>

pid_t getpgrp(void);

pid_t getpgid(pid_t pid);
```

Returns: group-ID; -1 on error (getpgid(2) only)

- in addition to having a PID, each process also belongs to a process group (a collection of processes associated with the same job/terminal)
- each process group has a unique process group ID
- process group IDs (like PIDs) are positive integers and can be stored in a `pid_t` data type
- each process group can have a process group leader
  - leader is identified by its process group ID == PID
  - leader can create a new process group, create processes in the group
- a process can set its (or its children's) process group using `setpgid(2)`

Process Groups

$

Process Groups


```
$ proc1 | proc2 &
[1] 10306
$
```

Process Groups


```
$ proc1 | proc2 &
[1] 10306
$ proc3 | proc4 | proc5
```

Sessions

```
#include <unistd.h>

pid_t setsid(void)
```

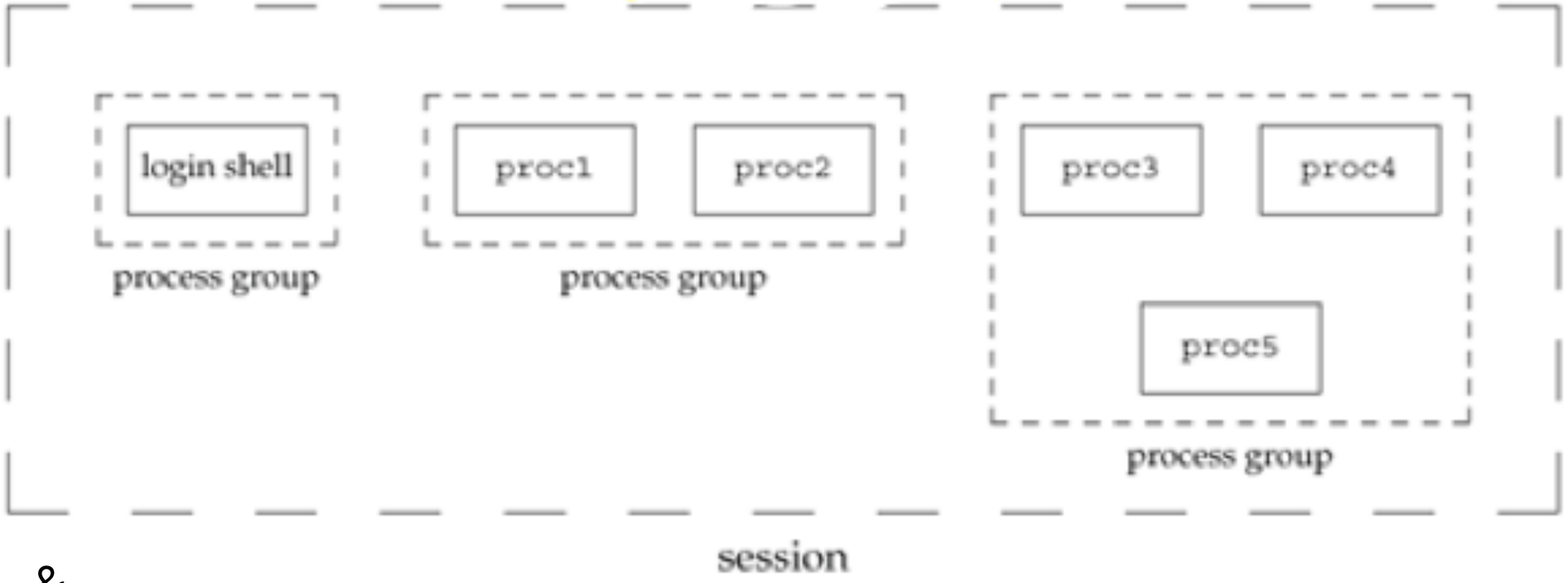
Returns: process group-ID if ok, -1 otherwise

A session is a collection of one or more process groups.

If the calling process is not a process group leader, this function creates a new session. Three things happen:

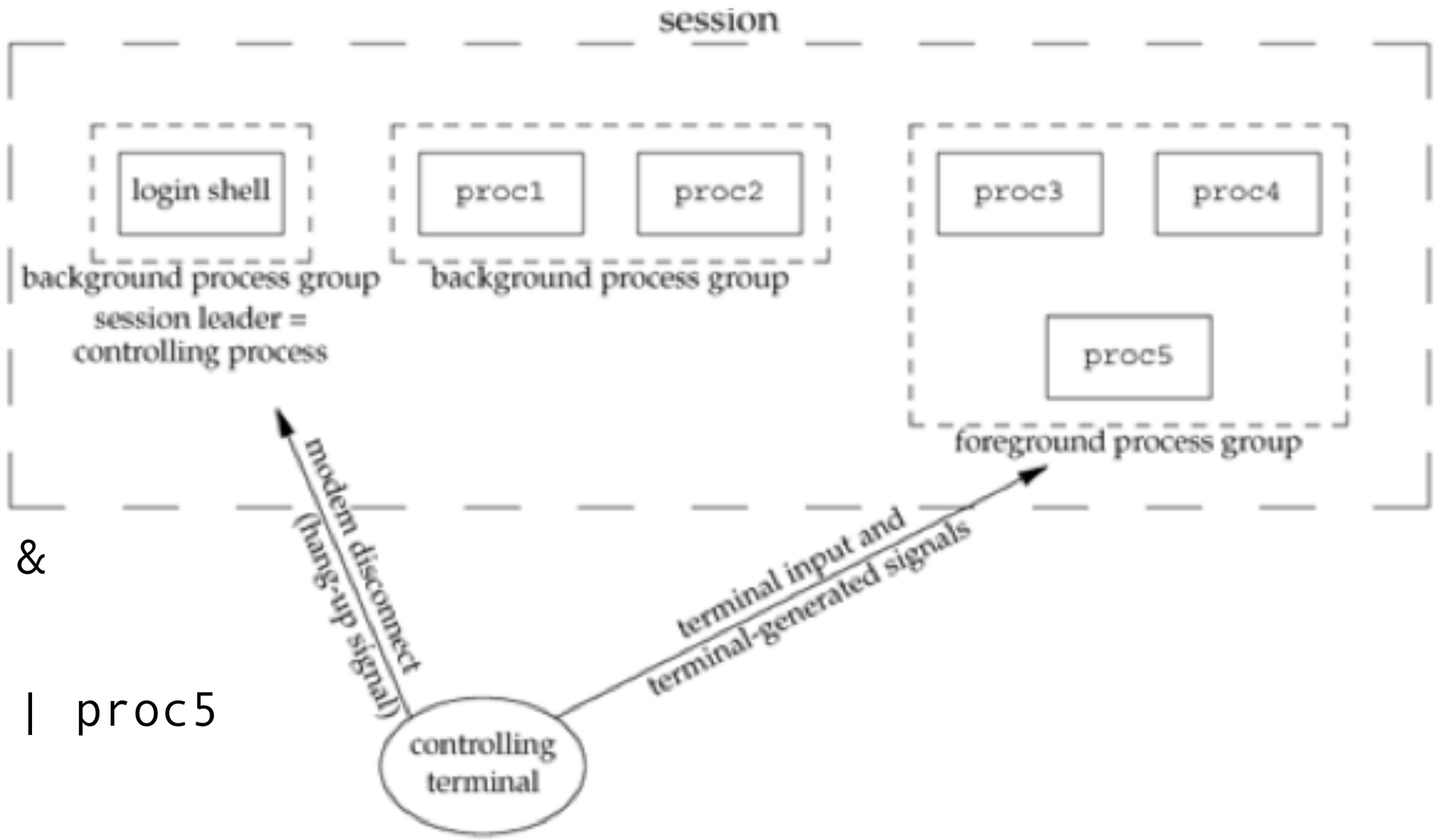
- the process becomes the session leader of this new session
- the process becomes the process group leader of a new process group
- the process has no controlling terminal

Sessions




```
$ proc1 | proc2 &
[1] 10306
$ proc3 | proc4 | proc5
```

Process Groups




```
$ proc1 | proc2 &
[1] 10306
$ proc3 | proc4 | proc5
```

```
apue$ ps -o pid,ppid,pgid,sid,comm
 PID PPID PGID  SID COMMAND
1579 1463 1579 1579 -ksh
1595 1579 1595 1579 tmux: client (/tmp/tmux-1000/default)
 471 1778  471 1778 ps
1778 1594 1778 1778 -ksh
1654 1594 1654 1654 -ksh
apue$ echo $$
1778
apue$ ps -o pid,ppid,pgid,sid,comm | egrep -v "(1778|1579)"
 PID PPID PGID  SID COMMAND
1654 1594 1654 1654 -ksh
apue$
```

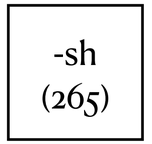
```
apue$ ps -o pid,ppid,pgid,sid,comm | egrep -v "(1778|1579)"
 PID PPID PGID  SID COMMAND
1324 1654 1615 1654 proc2
1615 1654 1615 1654 proc1
1654 1594 1654 1654 -ksh
apue$ ps -o pid,ppid,pgid,sid,comm | egrep -v "(1778|1579)"
 PID PPID PGID  SID COMMAND
1324 1654 1615 1654 proc2
1615 1654 1615 1654 proc1
1654 1594 1654 1654 -ksh
1705 1654 1705 1654 proc3
1709 1654 1705 1654 proc4
1844 1654 1705 1654 proc5
apue$ ps -o pid,ppid,pgid,sid,comm
 PID PPID PGID  SID COMMAND
1579 1463 1579 1579 -ksh
1595 1579 1595 1579 tmux: client (/tmp/tmux-1000/default)
1442 1778 1442 1778 ps
1778 1594 1778 1778 -ksh
1324 1654 1615 1654 proc2
1615 1654 1615 1654 proc1
1654 1594 1654 1654 -ksh
1705 1654 1705 1654 proc3
1709 1654 1705 1654 proc4
1844 1654 1705 1654 proc5
```

Process Groups

```
$ ps -o pid,ppid,pgid,sid,comm | ./cat1 | ./cat2
```

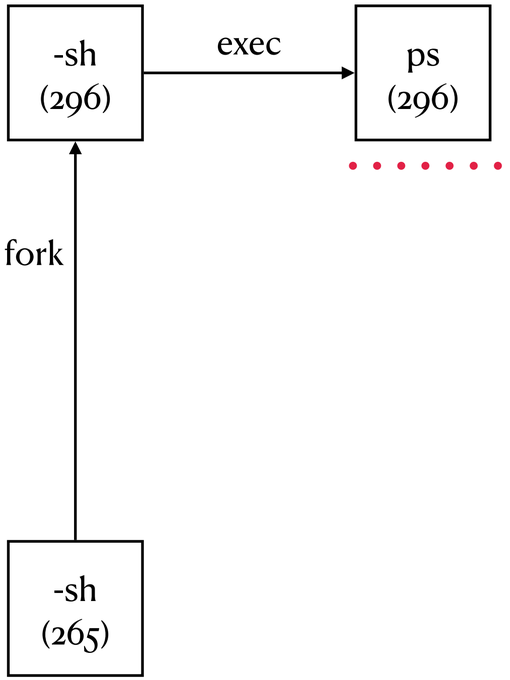
Process Groups

```
$ ps -o pid,ppid,pgid,sid,comm | ./cat1 | ./cat2
PID PPID PGID SID COMMAND
265  586  265 265 -sh
```



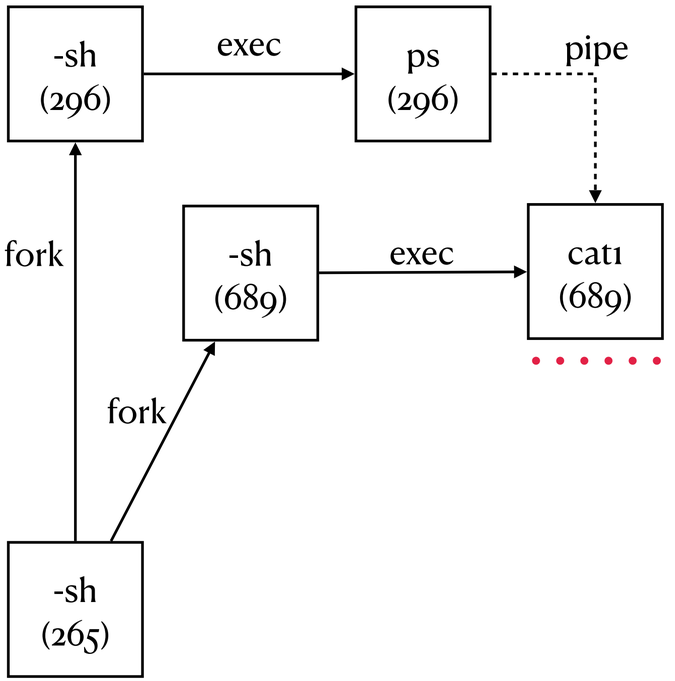
Process Groups

```
$ ps -o pid,ppid,pgid,sid,comm | ./cat1 | ./cat2
PID PPID PGID SID COMMAND
265  586  265 265 -sh
296  265  296 265 ps
```



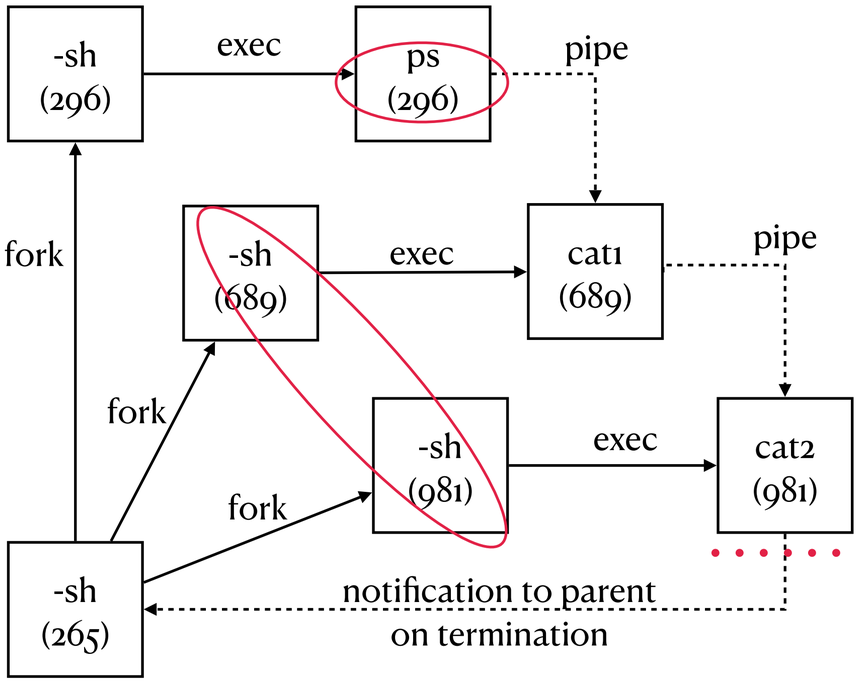
Process Groups

```
$ ps -o pid,ppid,pgid,sid,comm | ./cat1 | ./cat2
PID PPID PGID SID COMMAND
265  586  265 265 -sh
296  265  296 265 ps
689  265  296 265 -sh
```



Process Groups

```
$ ps -o pid,ppid,pgid,sid,comm | ./cat1 | ./cat2
PID PPID PGID SID COMMAND
265  586  265 265 -sh
296  265  296 265 ps
689  265  296 265 -sh
981  265  296 265 -sh
$ echo $$
265
```



Process Groups and Sessions

- each process belongs to a process group
- a session is a collection of one or more process groups
- process groups are used for distribution of (keyboard generated) signals

- process groups are used to implement job control in a shell:
  - processes that have the same process group as the terminal are foreground and may read
  - more on job control and signals in our next videos