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

```c
#include <unistd.h>
pid_t getpgrp(void);
pid_t getpgid(pid_t pid);
```

- 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

$
$ proc1 | proc2 &
[1] 10306
$
Process Groups

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

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

#include <unistd.h>

pid_t setsid(void)

Returns: process group-ID if ok, -1 otherwise
$ proc1 | proc2 &
[1] 10306
$ proc3 | proc4 | proc5
Process Groups

$ proc1 | proc2 &
[1] 10306
$ proc3 | proc4 | proc5
237 662 237 237 -sh
869 237 869 237 screen
843 290 843 843 /bin/sh
989 290 989 989 /bin/sh
1087 989 1087 989 ps
apue$ echo $$
989
apue$ ps -o pid,ppid,pgid,sid,comm | egrep -v "(989|237)"
  PID  PPID  PGID  SID   COMMAND
  843  290   843  843 /bin/sh
apue$ ps -o pid,ppid,pgid,sid,comm | egrep -v "(989|237)"
  PID  PPID  PGID  SID   COMMAND
  843  290   843  843 /bin/sh
  947  843   947  843 proc1
  985  843   947  843 proc2
apue$ ps -o pid,ppid,pgid,sid,comm | egrep -v "(989|237)"
  PID  PPID  PGID  SID   COMMAND
  843  290   843  843 /bin/sh
  891  843   891  843 proc3
  903  843   891  843 proc4
  947  843   947  843 proc1
  985  843   947  843 proc2
 1119  843   891  843 proc5
apue$
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
```

- `sh` (265)
Process Groups

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

<table>
<thead>
<tr>
<th>PID</th>
<th>PPID</th>
<th>PGID</th>
<th>SID</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>265</td>
<td>586</td>
<td>265</td>
<td>265</td>
<td>-sh</td>
</tr>
<tr>
<td>296</td>
<td>265</td>
<td>296</td>
<td>265</td>
<td>ps</td>
</tr>
</tbody>
</table>

fork

exec

-ps (296)

-sh (265)
$ ps -o pid,ppid,pgid,sid,comm | ./cat1 | ./cat2

<table>
<thead>
<tr>
<th>PID</th>
<th>PPID</th>
<th>PGID</th>
<th>SID</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>265</td>
<td>586</td>
<td>265</td>
<td>265</td>
<td>-sh</td>
</tr>
<tr>
<td>296</td>
<td>265</td>
<td>296</td>
<td>265</td>
<td>ps</td>
</tr>
<tr>
<td>689</td>
<td>265</td>
<td>296</td>
<td>265</td>
<td>-sh</td>
</tr>
</tbody>
</table>

---

**Process Groups**

```plaintext
fork

```

```plaintext
-exec

```

```plaintext
pipe

```

```plaintext
-fork

```

```plaintext
-exec

```

```plaintext
-cat

```

```plaintext
-ps

```

```plaintext
-sh

```

---

**Diagram:**

```
fork

```

```
-exec

```

```
pipe

```

```
-fork

```

```
-exec

```

```
-cat

```

```
-sh

```

---

**Diagram:**

```
fork

```

```
-exec

```

```
pipe

```

```
-fork

```

```
-exec

```

```
-cat

```

```
-sh

```
Process Groups

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

<table>
<thead>
<tr>
<th>PID</th>
<th>PPID</th>
<th>PGID</th>
<th>SID</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>265</td>
<td>586</td>
<td>265</td>
<td>265</td>
<td>-sh</td>
</tr>
<tr>
<td>296</td>
<td>265</td>
<td>296</td>
<td>265</td>
<td>ps</td>
</tr>
<tr>
<td>689</td>
<td>265</td>
<td>296</td>
<td>265</td>
<td>-sh</td>
</tr>
<tr>
<td>981</td>
<td>265</td>
<td>296</td>
<td>265</td>
<td>-sh</td>
</tr>
</tbody>
</table>

$ 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