

PROCESS GROUPS AND SESSIONS

Process groups

- A process group is a collection of one or more processes.
- Each process group has a unique process group ID.
- Process group IDs are similar to process IDs---they are positive integers and they can be stored in a `pid_t` data type.
- The function `getpgrp` returns the process group ID of the calling process.
- Each process group can have a process leader. The leader is identified by having its process group ID equal its process ID.

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

```
#include <unistd.h>
```

```
pid_t getpgrp (void);
```

- It is possible for a process group leader to create a process group, create processes in the group, and then terminate.
- The process group still exists, as long as there is at least one process in the group, regardless whether the group leader terminates or not
- process group lifetime — the period of time that begins when the group is created and ends when the last process in the group leaves the group
- A process joins an existing process group, or creates a new process group by calling `setpgid`.

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

```
#include <unistd.h>
```

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

- This sets the process group ID to *pgid* of the process *pid*. If the two arguments are equal, the process specified by *pid* becomes a process group leader.
- A process can set the process group ID of only itself or one of its children. If *pid* is 0, the process ID of the caller is used. Also if *pgid* is 0, the process ID specified by *pid* is used as the process group ID.
- In most job-control shells this function is called after a fork to have the parent set the process group ID of the child, and to have the child set its own process group ID.

SESSIONS

- A Session is a collection of one or more groups.
- The processes in a process group are usually grouped together into a process group by a shell pipeline.
- A process establishes a new session by calling the *setsid* function.

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

```
#include <unistd.h>
```

```
pid_t setsid (void)
```

- If the calling process is not a process group leader, this function creates a new session. Three things happen:
 1. The process becomes the session leader of this new session.
 2. The process becomes the process group leader of a new process group. The new process group ID is the process ID of the calling process.
 3. The process has no controlling terminal.

Controlling terminal

- characteristics of sessions and process groups
- A session can have a single controlling terminal.
- The session leader that establishes the connection to the controlling terminal is called the controlling process.

- The process groups within a session can be divided into a single foreground process group and one or more background process groups.
- If a session has a controlling terminal, then it has a single foreground process group, and all other process groups in the session are background process groups.
- Whenever we type our terminal's interrupt key or quit key this causes either the interrupt signal or the quit signal to be sent to all processes in the foreground process group.
- If a modem disconnect is detected by the terminal interface, the hang-up signal is sent to the controlling process

Source : <http://elearningatria.files.wordpress.com/2013/10/cse-iv-unix-and-shell-programming-10cs44-notes.pdf>