1. Process group

A group command on the shell makes a process group. The following three command is in the same process group.

# ls –l | grep “my.txt” | ps –e

Each group has leader process of group.

The process group is end when the last process in this group is terminal or it joins other groups.

Types: foreground and background process group

Function:

getpgrp: get the process group id ( leader process pid)   
setpgid: join a group or create a new group

Process only can set its own pid or the child pid( can not the one runs exec) to the group id

1. Session

A login makes out a session which is bound to a consoler.

There are several process groups in one session. But only one foreground process group and others are background. The signal and input from the related consoler will pass to all the processes in the foreground group.

Each session has a session leader which makes connection with the consoler.

Function:

pid\_t setsid: create a new session

The caller makes the process in the new session. The caller should not ay leader process of group. The caller becomes a leader of new process group and the session leader.

1. Jobs control

Jobs: used to control multiple process groups in a seesion.

1. jobs - List all the jobs that are running or suspended.
2. fg - Bring the job to the foreground. If it is in suspend status, resume it firstly.
3. bg - Send the job to the background. If it is in suspend status, resume it firstly.
4. kill or Ctrl + z - Suspend the job.
5. kill or Ctrl + c - Terminate the job.

You can point out a particular job using special identifiers. They are as follows.

%n - Job whose job number / job ID is the number n.

%+ - Current job. You can also use %%.

%- - Previous job.

How to manage jobs in Linux can be easily understood via an example.

Let’s simultaneously run a few commands that takes some time to complete

$ sleep 150 &

$ sleep 250 &

$ sleep 300 &

$ sleep 200 &

Here we have executed 4 sleep commands and all are started in the background as denoted by &.

List the running jobs

$ jobs

And the output is ...

[1] running sleep 150

[2] running sleep 250

[3] - running sleep 300

[4] + running sleep 200

In the output above, the number within [ and ] is the job number (job ID). The job number is unique for each job.

Bring a job to the foreground. If it is in suspend status, resume it firstly.

$ fg %job-number

To bring job number 2 to the foreground, you run the following command.

$ fg %2

Suspend the job

To suspend a job, you first bring the job to the foreground and then press the keys Ctrl + z.

Alternately, you can also suspend a job running in the background as follows.

$ kill -s SIGSTOP %2

Send a job to the background , If it is in suspend status, resume it firstly.

$ bg %job-number

To resume the suspended job 2 in the background, you can run the following command.

$ bg %2

List only running jobs

$ jobs –r

List only suspended jobs

$ jobs –s

List process IDs of the jobs

This command will list process IDs of the jobs in addition to the normal information.

$ jobs –l

Terminate a job

To terminate a job, you first bring the running job to the foreground, and then press the keys Ctrl + c.

You can also terminate a job without bringing it in the foreground just by passing the job number to kill.

$ kill %job-number

For example, to kill the 3rd job, you can do as follows.

$ kill %3