Job Control & Process Management

Introduction to Unix

Contents

Process

- An instance of running program
- PID : process identifier (1 – 32768)
- PPID : parent PID
- pid 0 : schedule demon
- pid 1 : init process
- $ ps [–l] // see (my own) process
- $ ps –ef // see all process
- $ echo $$ // see current shell pid

Shell Scripts

- # : comments
- #! : sh-bang
  - Tell the kernel to run the program listed after the #!
- my.sh
  - #!/bin/cat
  - Hello World
  - ^d
- chmod u+x my.sh
- ./my.sh

Stopping process

- Ctrl + C (^C)
  - Stop current process
- kill commands
  - Send SIGNAL to process
- kill –l
  - List signal names
- kill -9 pid
  - Kill –KILL pid
- pkill program_name
  - $ pkill sleep

/proc file system

- Contains a directory entry for active process named after PID
- $ ls –l /proc/$$
- $ ls –l /proc/1

SETUID

- When a regular user runs a program that are set a SETUID bit, the effective UID is changed to the UID of the program owner
- /usr/bin/passwd
- /usr/bin/crontab
- $ find / -perm -4000
- $ chmod u+s myprog

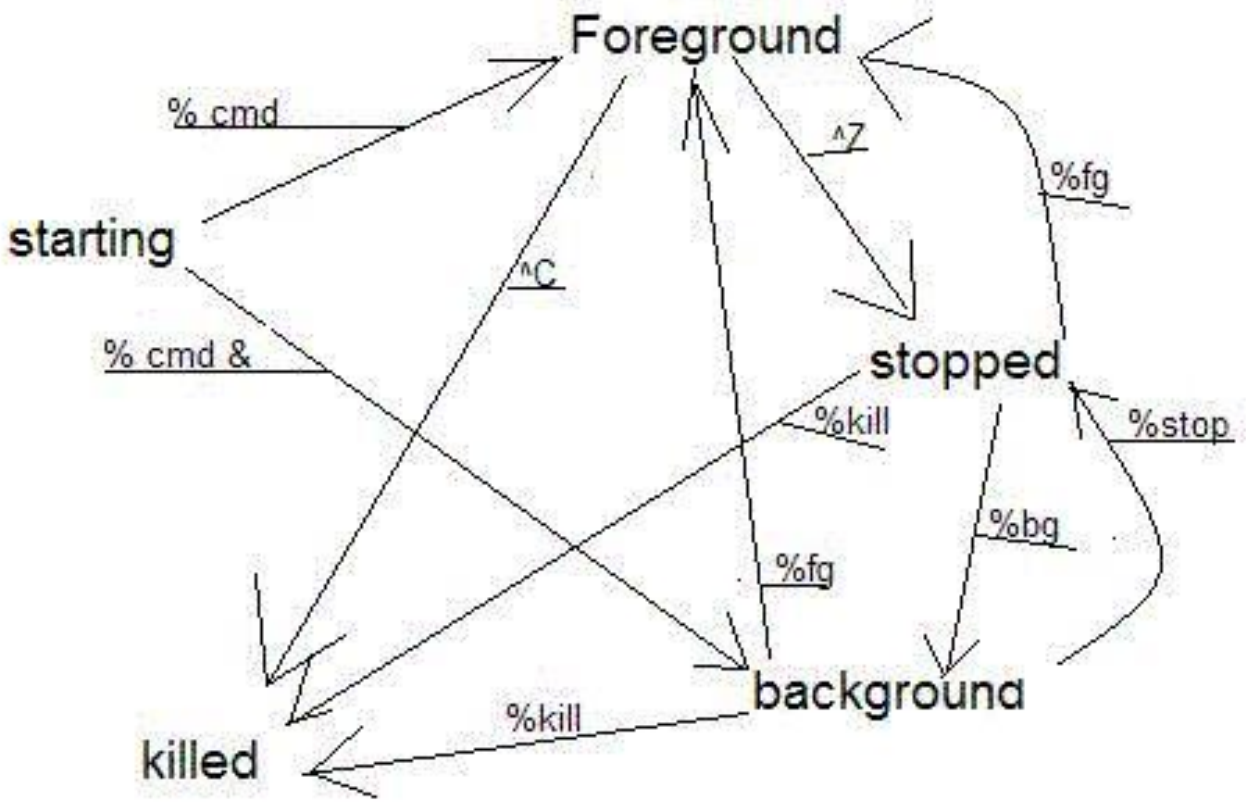
SETGID

- When a regular user runs a program that are set a SETUID bit, the effective GID is changed to the GID of the program owner
- /usr/bin/mail
- $ find / -perm -2000
- $ chmod g+s myprog

Job Control

- Two type of process
  - Foreground : seize the terminal
  - Background : release the terminal
- & (ampersand)
  - Start as a background
  - $ sleep 60 &
- $ jobs
- $ ps

Job control (csh)



Job commands

- Jobs // list jobs
- Fg [%n] // change to foreground
- Bg [%n] // change to background
- Kill %n // kill process
- Kill -18 %n // stop process
- Kill -19 %n // resume process
- Kill -9 pid // kill process