



CHALMERS



GÖTEBORGS UNIVERSITET

Introduction to Linux

Lecture 3

Prof. Dr. Philippe Tassin

Chalmers University of Technology

Dr. Evangelos Siminos

Chalmers University of Technology and University of Gothenburg

Göteborg, 10 April 2019

Processes

- “Everything in Linux is a file; if it is not a file then it is a process.”
- A program is a set of computer instructions;
a process is an instance of a program in action.

Interactive processes, automatic processes and daemons

- Interactive processes: started by someone at a terminal
 - Foreground or background
 - Foreground process has exclusive control over input
 - Job control commands: `fg`, `bg`, `jobs`
- Automatic processes
 - Not connected to a terminal
 - Started by `at` or `batch` or the cron system
- Daemons
 - Processes started by the system
 - Often during the boot procedure

Process attributes

- Name
- PID: every process has a *unique* number
- PPID: parent process
- Terminal (TTY)
- RUID: user ID of the user who started the process
- EUID: user ID used to determine access to system resources
(when SUID bit is set on executable)
- RGID
- EGID
- Environment variables
- File descriptors

Monitoring processes

- `top`: provides a real-time overview of processes
- `ps`: prints list of processes
- `pstree` or `pstree PID`
- `/proc/PID`: lots of information about every process

Fork-and-exec mechanism

- On Linux, new processes are created by cloning the parent process and then replacing the copy by another executable
- Child has same environment as parent (which can then be modified)
- Try `exec ls` in a terminal. What happens?

Boot process and run levels

- Kernel is loaded first
- Kernel loads `init` (PID=1)
- `init` takes care of system initialization
- `/etc/inittab`
- Then `init` starts daemons (SysVinit)
 - Daemons are started from scripts in `/etc/init.d`
 - Symbolic links in directories `/etc/init.d` start and stop services based on run levels

```
[jean@blub /etc/rc5.d] ls
K15httpd@      K45named@      S08ipchains@   S25netfs@      S85gpm@
K16rarpd@      K46radvd@      S08iptables@   S26apmd@       S90crond@
K20nfs@        K61ldap@       S09isdn@       S28autofs@     S90xfs@
K20rstatd@     K65identd@     S10network@    S30nscd@       S95anacron@
K20rusersd@    K74ntpd@       S12syslog@     S55sshd@       S95atd@
K20rwalld@     K74ypserv@     S13portmap@    S56rawdevices@ S97rhnsd@
K20rwhod@      K74ypxfrd@     S14nfslock@    S56xinetd@     S99local@
K25squid@      K89bcm5820@    S17keytable@   S60lpd@
K34yppasswdd@  S05kudzu@      S20random@     S80sendmail@
```

Systemd vs SysV init

- SysVinit has a number of disadvantages (slow; static dependency handling, static process handling)
- SysVinit is getting replaced (some controversy) by Systemd
- PID=1 process is now systemd
- Services described by unit files in */usr/lib/systemd/system* *and/or* */etc/systemd/system*
- Unit files contain information about how to start the service, about dependencies, etc.

Stopping processes

- Send signals:

Signal name	Signal number	Meaning
SIGTERM	15	Terminate the process in an orderly way.
SIGINT	2	Interrupt the process. A process can ignore this signal.
SIGKILL	9	Interrupt the process. A process can not ignore this signal.
SIGHUP	1	For daemons: reread the configuration file.

- Foreground process
 - Ctrl-C
- Other processes
 - Send signals with `kill`
 - `kill PID`
 - `kill -9 PID`

Scheduling processes

- Scheduling = make a process start at a time you want
- Four possibilities:
 - sleep
 - at
 - batch
 - cron
- sleep is a command that wait for a given number of seconds
 - (`sleep 60 ; echo "Print this after 1 minute"`)
 - mostly used in scripts
 - lazy but useful way to make sure some other task is completed

Scheduling processes

- `at`
 - Executes a command at a specified time
 - `at teatime - 1 hour Apr 4`
 - Fairly complex time definitions
 - To execute a command once
- `cron`
 - Executes a task repeatedly (e.g., weekly)
 - Configuration stored in a crontab file
 - `crontab -l`: show the configuration file
 - `crontab -e`: edit the configuration file

Any other questions?