GNU/Linux System Administration
Sketchy lecture notes

Marco Solieri

October 15, 2014

Lecture I

October 1st

1 Introduction

1.1 About the course

Lectures (*cours*):

- 5 sessions, 3h each
- please interact asking questions or commenting.

Practices (*travaux pratique*):

- 7 sessions, 3h each
- pair work on your server

Assessment

- theory side: written test
- practical side: *TP noté* on the 6th and 7th session

Bibliography

ULSAH Evi Nemeth, Garth Snyder, Trent R. Hein, and Ben Whaley, *Unix and Linux System Administration Handbook - 4th edition*. Prentice Hall (2010). http://ms.xt3.it/teach/sysadm/ulsah.pdf

TDAH Raphaël Hertzog, and Roland Mas,

- *The Debian Administrator's Handbook* http://debian-handbook.info/browse/stable/
- *Le cahier de l'administrateur Debian* http://debian-handbook.info/browse/fr-FR/stable/

(2013).

1.2 System administrator

System administrator duties:

- Account provisioning
- Adding and removing hardware
- Performing backups
- Installing and upgrading software
- Monitoring the system
- Troubleshooting
- Maintaining local documentation
- Monitoring security
- User assistance

You

- Who wants to be a SysAdm?
- Who is a SysAdm?
- How much do you already know about it?

1.3 Operating system

A layered abstractive view.

1.3.1 Kernel

The hardware interface and security guardian

CPU program execution, process management, pre-emptive multitasking, scheduler

Volatile memory RAM access control, virtual memory mapping, shared libraries

Permanent storage drivers for optical units, HDs and solid memory (IDE/PATA-SATA, SCSI, USB), filesystem abstraction

Networking network adapter drivers, TCP/IP stack (port management, firewall, routing)

1.3.2 System shell

The kernel interface

Command-line or graphical

File management, user management, process management, system configuration

1.3.3 Applications

Software execution: editors, interpreters, compilers, libraries

1.4 Unix

First wrote 1973 in Assembly and then in C

Design principles: portable, multi-tasking and multi-user in a time-sharing configuration.

Philosophy: plain text for storing data; hierarchical file system; devices and certain types of inter-process communication (IPC) as files; large number of software tools vs single monolithic program with many functionalities ("the idea that the power of a system comes more from the relationships among programs than from the programs themselves" – Kernighan and Pike).

Notable implementations:

- UNIX System V, IBM AIX, HP-UX, Mac OS-X, GNU/Linux ...
- BSD, FreeBSD, NetBSD, OpenSolaris ...

1.5 POSIX standard

IEEE standard, 1988

1.5.1 Core services

Process Creation and Control, Signals, Floating Point Exceptions, Segmentation / Memory Violations, Illegal Instructions, Bus Errors, Timers, File and Directory Operations, Pipes, C Library (Standard C), I/O Port Interface and Control, Process Triggers

1.5.2 Scheduling and concurrency

Priority Scheduling, Real-Time Signals, Clocks and Timers, Semaphores, Message Passing, Shared Memory, Asynch and Synch I/O, Memory Locking Interface

1.5.3 Threads

Thread Creation, Control, and Cleanup, Thread Scheduling, Thread Synchronization, Signal Handling

1.5.4 Shell and Utilities

Command Interpreter, Utility Programs

1.6 Other standards

1.6.1 Executable and Linkable Format

De facto standard (around 1997 in SysV)

File format for executables, object code, shared libraries, and core dumps.

1.6.2 Filesystem Hierarchy Standard

BSD derived, 1994 and ongoing

Directory structure and directory contents

1.6.3 Linux Standard Base

Ideal standard, not fully used

Standardize the software system structure, extending POSIX and FHS

standard libraries, commands and utilities, file system hierarchy, run levels, printing system (CUPS, Foomatic), extensions to the X Window System.

1.7 GNU/Linux

1.7.1 GNU

GNU's not Unix project starts in 1983, by RMS

GNU software compilation:

- base tools: coreutils, grub, tar, gzip ...
- use tools: glibc, common lisp, make, GPG, Gnome, Mailman, R, Octave ...
- kernel: ? Hurd still in development

1.7.2 Linux

The Linux kernel in 1991, by Linux Torvalds

Derived by MINIX

1.7.3 GNU/Linux distributions

Public available repository: software available for immediate installation

Dependency management between programs and libraries and their versions

Tradeoff between easiness and control in system administration: lot of preconfigured packages and administration tools vs vanilla sources

Free software: adaptable, transparent, secure

1.7.4 Debian GNU/Linux

Debian, the universal operating system:

- sweet spot between easy and light
- community-driven
- free and non-free software separated
- dozen of derivatives (Ubuntu)

1.8 Documentation

1.8.1 Commands

help help information about shell built-in

man manual pages (text only)

info texinfo pages (hypertext)

apropos search man pages

which search binary of a command

whereis search specific files about a command (binaries, libraries, configuration files, sources, ...)

find search for files

locate search for file in a database

1.8.2 Software

- standard documentation /usr/share/local
- text, hypertext, hypertext with graphics

1.8.3 Task

- HowTos on blogs
- StackExchange

1.8.4 Distribution

wiki, manual books, bug reports

1.8.5 General

newsgroup, mailing lists, forums, IRC channels

2 System setup and configuration, an overview

2.1 Installation of a GNU/Linux distribution

2.1.1 Bootstrap

CD or USB drive: just burn or `dd`

Network: with Preboot eXecution Environment PXE (DHCP server, Trivial File Transfer Protocol (TFTP) server)

2.1.2 Base hardware configuration

Base hardware recognition: mostly automatic, sometimes you need to load additional drivers for network cards

Disk setup: partitioning for system files, swap (paging) space,

2.1.3 Base system installation and configuration

Installation of base system: kernel, libraries, tools; check for updates

Users: the administrator `root` and simple user, or just the latter with ability to become administrator with `sudo`.

Installation of the boot loader GRUB; able to recognize and boot other OSes installed.

3 Basic security and access control

3.1 Users and groups

Aim: control access to resources

User:

- ID,

- name,
- information,
- password,
- expire date,
- home directory,
- shell (/bin/false)

Group:

- ID,
- set of users.

Configuration files:

- /etc/passwd
- /etc/shadow
- /etc/groups

3.2 Users and groups management

- Addition: user add/adduser, groupadd/addgroup
- Removal: userdel/deluser
- Configuration: usermod, groupmod
- Locking:
  - lock the password: passwd -l (-u to unlock)
    Check if key login available!
  - set an expire date in the past

3.3 Basic access control management

File access permissions. Who can do what?

Nine bits matrix:

Entities user, group, others.

Actions on files (or dirs)
- read (or list files),
- write (or add/remove files),
- execute (or traverse).

Default permissions in file creation: umask.

3.3.1 Execute with different rights

- su (substitute user): default root, other users can be specified
- sudo: execute as root (then you can disable root access at all)
- newgrp: set another group as temporary default

References

ULSAH: §1, §4, §7,

TDAH: §4, §8.4, §8.5.

Lecture II

October 7th

4 System boot and services

References: §3 ULSAH

4.1 Boot

4.1.1 Boot loader (GNU GRUB)

- config in /boot (some is generated from /etc/grub... at each update)
- loader installed in the MBR (`grub-install`)
- menu for selection and shell interface
- select disk partition containing the root
- execute the kernel with parameters, e.g.
  - `init=/bin/bash` for emergency recovery
  - `single` single user mode
  - `quiet`

4.1.2 Linux

- Kernel images are not replaced during update
- initramfs
- custom configure (and patch) and build
- modules: dynamical load or built-in

4.2 Init

Common kernel processes:

kjournald Commits filesystem journal updates to disk

kswapd Swaps processes when physical memory is low

ksoftirqd Handles soft interrupts if they can't be dealt with at context switch time

Init

- PID 1
- Operator intervention (single user flag): enter root's password or, CTRL-D to continue.
- Determines which services should be started.

Runlevels

- base values: 0=halt, 1=single user, 6=reboot
- default levels
  - 2: Debian;
  - 3: Gentoo, Slackware, Arch;
  - 3/5: RedHat console/graphical.

Service management scripts

- Location:
  - master copy in `/etc/init.d/`
  - `/etc/rc$RUNLEVEL.d/$SK$SEQN$NAME`
- Default run level behaviour determined by `rc`
  - `$SK` for Start or Kill
  - `$SEQN` for ordering
- Manual actions invocating the script with arguments start, stop, status, restart, reload.

New systems enjoying parallel start

- Systemd
- Upstart

5 Process management

References: §5 ULSAH

Recall: process and thread

PID

Parent (process tree)

Process status: waiting, running, zombie ...

Priority: nice and IO nice

Management:
signals 15, 9
kill, killall

File descriptors: standard in/out/err

6 Basic software management

6.1 Software configuration

RC files: run control

Form: text, line-oriented, sysadm-readable

Location: systemwide in '/etc/', user '~/.dotfilerc'

DebConf tool: dpkg-reconfigure command, questions with priorities, different frontends (text, ncurses, Kde), database storage, generation of conf files.

7 Security and users administration

7.1 File access control

7.1.1 File attributes

Content
owner, group, creation date, modification date, access date, permissions

"hidden" dot files

7.1.2 Special permissions

setuid and setgid bit: execute as the owner user or group

sticky bit: only the item's owner, the directory's owner, or the superuser can rename or delete files

Lecture III

October 9th

8 Storage management

References: §8 ULSAH

8.1 Storage media

Size, costs, durability, access speed

- optical disc
- tape (DDS)
- hard disk
- SSD

Interfaces

- USB
- IDE-PATA, SATA
- SCSI, SAS
- fibre

8.2 Storage architectures

Partitions: DOS table and types (primary, extended, logical)

RAID: hardware/software, levels, performance, costs

LVM: physical volume, volume group, logical volume; extending, moving, mirroring

8.3 Filesystems

Examples: ext2, ext3, ext4, ReiserFS, Reiser4, XFS, ZFS.

Concepts: block, journaling, extents.

Tools: mkfs, chkfs.

8.4 Tools

SMART (Self-Monitoring, Analysis and Reporting Technology)

Test: HDParm

9 Filesystem management

9.1 Mounting

References: §8 ULSAH

Concept: make a storage media accessible in the existing filesystem.

Functioning: activate drivers, map the space into directory in `/`.

Automatic mounting

- at boot time with `/etc/fstab`
- at plug time with some daemon (e.g. `automount`)

Permission

- root user,
- non-root user, in user directories only, with `udisks` or other desktop tool (modern systems only).

Dynamic list of mounts:

- in `/etc/mtab` file
- in `mount` output

9.2 Filesystem Hierarchy Standard

`/` Primary hierarchy root and root directory of the entire file system hierarchy.

`/bin` Essential command binaries that need to be available in single user mode; for all users, e.g., cat, ls, cp.

`/boot` Boot loader files, e.g., kernels, initrd.

`/dev` Essential devices, e.g., /dev/null.

`/etc` Host-specific system-wide configuration files

`/etc/opt` Configuration files for /opt/.

`/etc/X11` Configuration files for the X Window System, version 11.

`/home` Users' home directories, containing saved files, personal settings, etc.

`/lib` Libraries essential for the binaries in /bin/ and /sbin/.

`/media` Mount points for removable media such as CD-ROMs (appeared in FHS-2.3).

`/mnt` Temporarily mounted filesystems.

`/opt` Optional application software packages.

`/proc` Virtual filesystem providing information about processes and kernel information as files. In Linux, corresponds to a procfs mount.

`/root` Home directory for the root user.

`/sbin` Essential system binaries, e.g., init, ip, mount.

`/srv` Site-specific data which are served by the system.

`/tmp` Temporary files (see also /var/tmp). Often not preserved between system reboots.

`/usr` Secondary hierarchy for read-only user data; contains the majority of (multi-)user utilities and applications.

`/usr/bin` Non-essential command binaries (not needed in single user mode); for all users.

`/usr/include` Standard include files.

`/usr/lib` Libraries for the binaries in /usr/bin/ and /usr/sbin/.

`/usr/local` Tertiary hierarchy for local data, specific to this host. Typically has further subdirectories, e.g., bin/, lib/, share/.

`/usr/sbin` Non-essential system binaries, e.g., daemons for various network-services.

`/usr/share` Architecture-independent (shared) data.

`/usr/src` Source code, e.g., the kernel source code with its header files.

`/usr/X11R6` X Window System, Version 11, Release 6.

`/var` Variable files—files whose content is expected to continually change during normal operation of the system—such as logs, spool files, and temporary e-mail files.

`/var/cache` Application cache data. Such data are locally generated as a result of time-consuming I/O or calculation. The application must be able to regenerate or restore the data. The cached files can be deleted without loss of data.

`/var/lib` State information. Persistent data modified by programs as they run, e.g., databases, packaging system metadata, etc.

`/var/lock` Lock files. Files keeping track of resources currently in use.

`/var/log` Log files. Various logs.

`/var/mail` Users' mailboxes.

`/var/run` Information about the running system since last boot, e.g., currently logged-in users and running daemons.

`/var/spool` Spool for tasks waiting to be processed, e.g., print queues and unread mail.

`/var/spool/mail` Deprecated location for users' mailboxes.

`/var/tmp` Temporary files to be preserved between reboots.

9.3 Device files

`fb` framebuffer

`fd` floppy

`lp` printer

`pt` pseudo terminal

`tty` terminal

`hd` old IDE/PATA drive (HD, CD, DVD...)

- units names: a,c,b,d
- primary partition names: 1-4
- logical names: 5-

`sd` SCSI, SATA, new PATA, USB, Firewire

- primary partition names: 1-4
- logical names: 5-

`null` accepts and discards all input; produces no output.

`zero` accepts and discards all input; produces a continuous stream of NULL (zero value) bytes.

`random` produces a variable-length stream of pseudo-random or truly random numbers. (Blocking)

`urandom` produces a variable-length stream of pseudo-random numbers. (Non-Blocking)

Lecture IV

October 10th

10 Software management

References: §12 ULSAH

10.1 Manual, from developers

A typical workflow:

1. Choose the version
   - stable vs latest?
   - find and install dependencies
2. Download the source
   - compressed archive from website or FTP
   - directory import from development repository (SVN/GIT/HG)
   - verify integrity (MD5 or SHA1) or authenticity (PGP signature)
3. Compile
   (a) configure scripts: operating system, processor type
   (b) manual editing or autoconf tools (.configure)
   (c) build make
4. Install
   manual or automatic (.install)
5. Configure
   - copy and edit example configuration files
   - read manuals
6. Maintainance
   - check for security update and repeat the whole process
   - check for major update and repeat the whole process (dependencies may break)

10.2 Aided, from distribution

References: §5, 6 TDAH

Key ingredients:

- repository: updated continuously
- binary packages: precompiled executables and libraries (compile-time conf fixed), documentation, configuration examples
- scripts for installation, removal, configuration, and removal processes
- source packages: like the bin, except you can change compile-time configuration

Benefits: software management process made easy (find software, install, uninstall, update, control).

Major examples and tools:

- Debian PMS (Debian and derivatives): `dpkg`, apt and aptitude, synaptic ...
- Red Hat PMS (RHEL, CentOS, Suse): rpm, yum, other software center

11 Scheduling

References: §9 ULSAH

11.1 Cron

Periodic job scheduler

Cron table (crontab) syntax:

```
 *    *    *      *     *    sh command to execute
min hour daym  month  dayw
```

where:

- min (0 - 59)
- hour (0 - 23)
- day of month (1 - 31)
- month (1 - 12)
- day of week (0 - 7) (0 or 7 are Sunday, or use names)

Also:

- , for separating sets of values
- /n for specifying intervals

Original syntax is extended by modern implementations.

Every user may have his crontab.

Predefined directories in `/etc/cron.*`:

- monthly,
- weekly,
- daily,
- hourly

11.2 Other tools

Non-periodic scheduling: `at`.

12 Logging

References: §11 ULSAH

12.1 Motivation

Goal: record activities and events of the server

Typical applications:

- troubleshooting
- security incident response
- security policies
- auditing policies
- forensics

12.2 Syslog

Centralised and standardised logging system

Two-dimensions categorisation:

facility
- auth,
- authpriv,
- daemon,
- cron,
- ftp,
- lpr,
- kern,
- mail,
- news,
- syslog,
- user,
- uucp,
- local0, . . . , local7

severity
1. Debug
2. Info,
3. Notice,
4. Warning,
5. Error,
6. Critical,
7. Alert,
8. Emergency,

Files stored in `/var/log/$FACILITY`.

Recorder `logger`.

Tools:

- log file rotation: `logrotate`,
- automated log analysis: `logcheck`.

13 Advanced topics in filesystem management

13.1 Access Control List

arbitrary length list of pairs: actor (user or group), permission (r,w,x)

directory inheritable lists

more control, more complexity

optionally available, with some differences between FS

13.2 Jailing

Creating a separate root filesystem (a jail) for:

- testing,
- new system installation,
- or honeypots.

How:

1. change root `chroot`
2. optional help for jail construction with `makejail`

Lecture V

October 15th

14 System backup and restore

References: §10 ULSAH

14.1 Goal

Preserve old versions of permanent storage

Define the time view in two dimensions:

- length,
- granularity (usually decreasing over time)

14.2 Principles

Separate physical locations of data and backup: HDs, machines, rooms, buildings, state, continent.

Data supports: CD, DVD, tapes, HDD, SSD,

Tapes:

- Magnetic data tape (DDS)
- Typical size: hundreds of GB
- Built-in compression
- Comparison wrt other media techonologies:
  - price: for large amount of data, the cheaper
  - lifetime: 15~30y vs 5~10y of CDs vs 5~20y of flash
  - hot-swappable
  - autoloadable in libraries

Space-usage optimisations techniques:

- Differentials/incremental backups: store only differences against the last full backup copy.
- Overlapping copies: store different copies of the same storage source (taken in different moments) in two directory trees where unchanged files are links to the same inode. (Works only with hard-link capable filesystems.)
- De-duplication: find duplicate files, possibly belonging to different storage units or even different hosts, and make them point to the same inode. (Works only with hard-link capable filesystems.)

Dedicate backup for DBs: copying DBMSs' files is not sufficient and not safe.

14.3 Examples

Archiving:

- dump and restore tools
- compressed archives: tar + compression (gzip, bzip, ...)

Transfer and store: burn to discs, write on tape, cp, FTP or SFTP (file put, Fuse mount) on the network to some fileserver.

File synchronizer: Unison, Rsync

Do not forget to secure your storage: no OS, no permissions!

Off the shelf solutions: BackupPC (rsync, CIFS), Bacula (tapes)

15 Security

15.1 Authentication

References: §22 ULSAH

PAM

Local login : password

Remote login (SSH): password or key

Password management: renewal, expiration

15.2 Advanced access control management

References: §6 ULSAH

Access Control List
optionally available

polkit (former PolicyKit): access control mechanism, finer than sudo

consolekit (now superseded by systemd): tracking users sessions

15.3 Mandatory access control (MAC)

References: §22 ULSAH

Main concept: separating control policy from access policy – he who has access cannot grant it.

Implementations:

- grsecurity
- Security Enhanced Linux (SELinux), used by NSA

Hard to configure, hard to attack

15.4 Advanced account management

NIS/YP

LDAP

16 Remote access

16.1 Command line

gpm for mouse in console (no cursor movement)
copy-paste with selection-right click

history navigation with arrows
ctrl- combination (P previous history, R reverse history search, M enter, ...)

terminal: tty, screen, graphical emulator

SSH: Secure Shell

- password authentication
- key: `.ssh/authorized_keys`

16.2 Remote access

Graphical environment architecture:

- X Window system: X server architecture
- Window manager
- Desktop environment: Gnome, KDE
- Display manager: login and choose session

SSH & X forwarding

- SSH connection with `-X` option, which automatically sets up everything for connecting the shell to an X server
- encrypted communication
- partial access: single applications

XDMCP

- two X servers running on both client and server hosts
- dedicated UDP protocol
- unencrypted connection, suitable in trusted (virtual) networks
- complete access: full desktop session on server is available on client

17 Monitoring

References: §11 ULSAH

Reporting: SAR, Munin

Watchdog: Monit

Intrusion Detection Systems