

OpenLab Virtual 2021

Matteo Savatteri

5 marzo 2021

Indice

1 GNU/Linux Intro, Tips & Tricks	3
1.1 Introduzione	3
1.2 Storia di GNU/Linux	3
1.3 GNU/Linux per un Fisico	5
1.4 Presentazioni ospiti	5
1.5 Chiusura	5
2 CLI Essential and Shell Programming	6
2.1 Nozioni e Strumenti	6
2.1.1 Setup	6
2.1.2 Il concetto di <i>Shell</i>	6
2.1.3 <i>Interprete dei Comandi</i> vs <i>Emulatore di Terminale</i>	7
2.1.4 bash	7
2.1.5 Programmi Applicativi di UNIX	8
2.2 Chiusura	10

Capitolo 1

GNU/Linux Intro, Tips & Tricks

1.1 Introduzione

- Benvenuto e ringraziamenti ad admin e pubblico
- Chi sono?
- Cos'è LCM?
- Regole conferenza: come interagire
- Outline e obiettivi conferenza:
 - Storia di GNU/Linux
 - GNU/Linux per un fisico
 - GNU/Linux: nozioni e strumenti importanti
 - Testimonianze ospiti + domande

1.2 Storia di GNU/Linux

- UNIX ¹ (*Uniplexed Information and Computing Service*, "eunuchs"), '69, Ken Thompson, Dennis Ritchie, Brian Kernighan & Co., Assembly:
 - Filosofia di UNIX: *"This is the Unix philosophy: Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that is a universal interface."* - Doug McIlroy
 - AT&T, Bell Labs

¹Video *AT&T Archives* introduzione a UNIX a <https://youtu.be/tc4R0CJYbm0>

- '69-73, sviluppo del C, K.T. & D.R. → Portabilità
- Distribuzione università (es. Berkeley, BSD) → Codice aperto
- '84 commercializzazione → Codice chiuso!
- Standard:
 - * '85 POSIX (Portable Operating System Interface for UNIX), IEEE
 - * '98 SUS (Single UNIX Specification), The Open Group
- GNU (*GNU's Not UNIX*), '84, Richard Stallman:
 - Sistema operativo Libero (cosa significa?) ² clone di UNIX
 - '90 Sviluppo di tutte le componenti del sistema
 - Esempi software GNU: `gcc`, `glibc`, `bash`, `coreutils`, GNOME, etc.
 - '90 inizia sviluppo GNU Hurd
- Digressione su OS:
 - OS = Insieme di programmi che consentono all'operatore di utilizzare un computer
 - Kernel = Nucleo dell'OS deputato ad allocare le risorse della macchina e altri servizi agli altri programmi
- Linux³, '91, Linus Torvalds:
 - Progetto personale annunciato su `com.os.minix`
 - Kernel Libero (GNU GPLv2), Monolitico, Modulare, UNIX-like
 - '92 Linux diventa kernel del sistema GNU
 - Adozione in ambito commerciale
 - Prima diffusione in supercomputer (Vedi TOP500 OS Family)
 - Poi leader in ambito server ⁴ (in particolare LAMP)
 - 2000, Linux Foudation → LSB (*Linux Standard Base*) Standardizzazione OS che adottano kernel Linux (Da tempo *standard de facto*)
 - Oggi diffuso in dispositivi mobili (**Android**), embedded (router board, wearables) e altre applicazioni (es. *TESLA*)
 - Percentuale diffusione workstation, desktop, laptop e gaming (es. Valve Corporation, SteamOS) in crescita
- Concetto ed esempi di distribuzione GNU/Linux (Ubuntu, Mint, Fedora, Arch, Gentoo, Debian, CentOS, RHEL, SUSE, LFS, OpenWRT, Raspbian); cosa distingue varie distro? → *package manager*, *init system*, configurazione kernel Linux, etc.

²Per approfondire vai a <https://www.fsf.org/it>

³Pronuncia del nome Linux a <https://mirrors.edge.kernel.org/pub/linux/kernel/SillySounds/>

⁴Guarda ad esempio statistiche su web server a https://w3techs.com/technologies/overview/operating_system

1.3 GNU/Linux per un Fisico

Perhé un fisico dovrebbe conoscere ed utilizzare il sistema GNU/Linux?

- Cluster di computer e farm (es. MARCONI, INDACO, LCM Farm) nel mondo sono quasi interamente costituite da reti di sistemi GNU/Linux (es. CentOS, RHEL, Debian)
- Codice libero = adattabilità, trasparenza, riproducibilità, no interessi
- Calcolo scientifico ha storia basata su sistemi UNIX-like da decenni → applicazioni vengono prevalentemente sviluppate su e per questa famiglia di sistemi
- Tutte le persone dovrebbero usare un OS Libero, per ragioni etiche

1.4 Presentazioni ospiti

- Ospiti: Zec, Sboz, Blue, Algebrato (?)
- 5 min a testa
- Domande

1.5 Chiusura

- Repo dell' *OpenLab*, cosa trovate, licenza
- Contatti LCM: sito, gruppo community, mail
- Saluti

Capitolo 2

CLI Essential and Shell Programming

2.1 Nozioni e Strumenti

2.1.1 Setup

- Vedete il terminale?
- Il font è abbastanza grande?
- Vedete il puntatore?
- Non andare troppo veloce
- Spiega che tasti premi, se non vengono printati
- Non cancellare l'output del terminale

2.1.2 Il concetto di *Shell*

- *shell* è programma per computer che espone all'utente o ad altri programmi i servizi dell'OS
- Si chiama così perché è strato più esterno dell'OS
- Può essere GUI (es. GNOME, KDE) o CLI (es. `sh`, `ksh`, `bash`, `csh`, `COMMAND.COM`)
- Nel caso di shell CLI → *interprete dei comandi*

2.1.3 *Interprete dei Comandi vs Emulatore di Terminale*

- *emulatore di terminale* = GUI che emula una *telescrivente* ¹
 - Programma *master* che riceve e interpreta le stringhe digitate dall'utente alla tastiera, le mostra in una finestra grafica e le passa al programma *slave* (es. *interprete dei comandi*)
 - Riceve stringhe dal programma *slave* e le mostra all'utente grazie ad una finestra grafica
 - Esempi: `xterm`, `urxvt`, `st`
 - Quando non è un GUI si chiama *console virtuale* (es. le Linux `tty`)
- *interprete dei comandi* = una shell di tipo CLI
 - Programma *slave* che riceve stringhe dal programma *master*, le interpreta e chiede a OS di eseguire programmi corrispondenti
 - Può passare stringhe al programma *master* perché vengano mostrate a schermo

2.1.4 `bash`

- Shell del progetto GNU
- `bash` = *Bourne Again SHell* (Stephen Bourne)
- È interprete dei comandi e linguaggio di scripting
- Se chiamata come `sh` o con la flag `-posix` comportamento conforme al POSIX (`sh`)
- Feature:
 - Esecuzione di comandi → Se no slash: ricerca → funzioni, builtin, PATH (`hash`) → Esecuzione con assegnazione argomenti
 - Shell scripts
 - Environment (`export`)
 - Simple commands → “A simple command is a sequence of optional variable assignments followed by blank-separated words and redirections, and terminated by a control operator.”
 - Pipeline → `|`, `|&`, in subshell
 - Liste di comandi → `;`, `&`, `&&`, `||`
 - Compound commands → `(list)`, `{ list; }`, `((expression))`, `if`, `for` (due tipi), `while`, `until`, `case`

¹Video interessante di *telescrivente* connessa a sistema GNU/Linux: <https://www.youtube.com/watch?v=9TGQ4pnVWSQ>

- Shell functions → `fname () compound-command [redirection]`
- Coprocesses → `coproc [NAME] command [redirections]`
- Commenti → `#`
- Quoting
- Parametri → “*A parameter is an entity that stores values.*”, “*A variable is a parameter denoted by a name.*”, definizione, valutazione, parametri speciali (`*`, `@`, `#`, `?`, `-`, `$`, `!` (job), `0`) e variabili speciali (in particolare `PATH`), namerefs (`declare -n ref=VAR`), `declare builtin`
- Array → `name=(value1 ... valuen)`
- Espansione → brace expansion, tilde expansion, parameter and variable expansion, command substitution, arithmetic expansion, word splitting, and pathname expansion
- Redirezione → `>`, `>>`, `<`, nomi file spaciali, *Here Documents* `[n]<[-]word ... delim`, *Here Strings* `[n] <<< word`, duplicating file descriptors `[n]>&word` (analogo per input, moving file descriptors `[n]>&digit-` (analogo input), opening file descriptors for reading and writing `[n]<>word`.
- `history`
- Tab completion
- Job control
- Builtins (Vedi man page)

2.1.5 Programmi Applicativi di UNIX

Il Comando Più Importante dell’Universo

Comando `man` è più importante universo! `man man man!` Vedi anche `apropos o man -k`.

Regole per documentarsi:

- Cercare nelle *man page*, `info`, `-help` e solo infine sul web
- Cercare *in lingua inglese* → È la lingua di internet e dell’informatica
- Cercare sempre dalla documentazione ufficiale → Meglio diffidare di guide e blog, se possibile. (*Chi l’ha scritto, è chi meglio ti può consigliare*)
- Vai a fondo finché il tempo a tua disposizione te lo permette

File System

- Digressione su VFS: radice, nodi, file regolari, file speciali (?)
- `pwd`
- `touch` → Puoi introdurre qui, :

- cd → No argomento, -
- ls → -l, -a, -t, -r
- cat → -et
- cp → -r, dir/., -
- mv → -i
- unlink
- rm → -r, -i, -f
- mkdir → -p
- file → -l
- df → -a, -i, -h
- du → -a, -h, s
- find → -name, -iname, combinazione con xargs
- cmp → diff, diff3
- dd
- ln → -s
- readlink

Text Processing

- cut → -f, -d
- paste → -d
- head → -N
- tail → -N, -f
- sort → -h, (tsort)
- wc → -l
- join → ?
- more
- less → keybind v
- sed → *Stream editor*, -i, s/pat1/pat2/g, {N}d
- tr → -d
- uniq
- nl → Numera linee file argomento

Misc

- `uname` → `-a`
- `date`
- `cal` → `-y`
- `bc`
- `split` → `-b` (bytes)
- `env`
- `grep` → `-c`, `-v`
- `id`, `whoami`, `who`, `logname`
- `od` → `-a`, `-x`
- `tty`

Shell Programming

- `echo` → `-e`, `-n`
- `sleep`
- `tee` → `-a`
- `command`
- `true`
- `false`

2.2 Chiusura

- Repo dell'*OpenLab*, cosa trovate, licenza
- Contatti LCM: sito, gruppo community, mail
- Saluti