



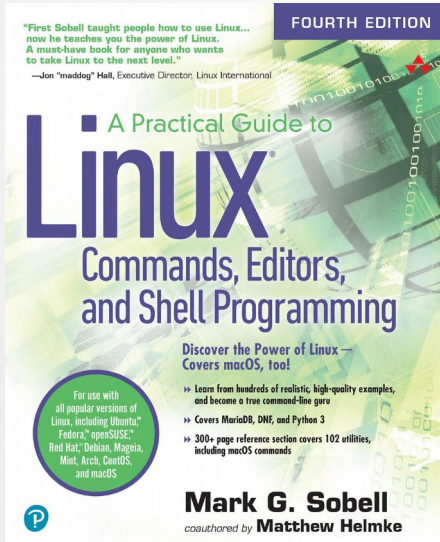
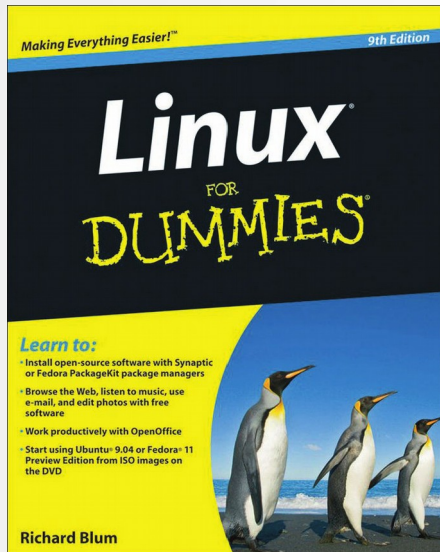
LINUX & Ubuntu

Uma breve introdução

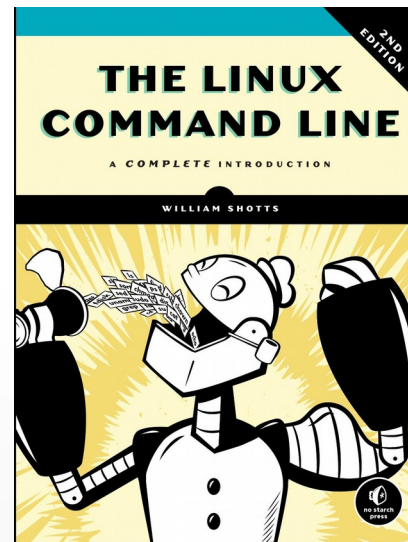
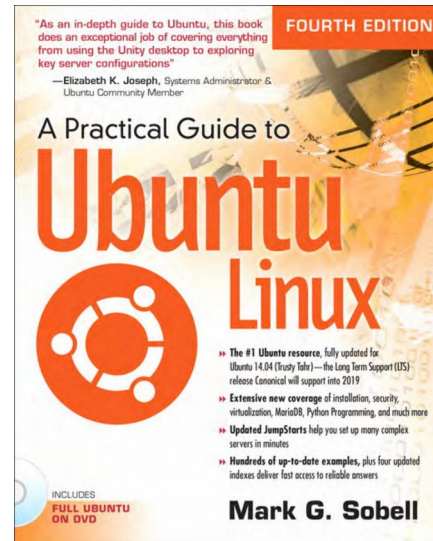
V. Santos, 2015 – 2019

Bibliografia sugerida

Iniciação



Guia mais completo



Focados na linha de comando

Linux... ubuntu – o que são?

- Linux - Sistema operativo
 - conjunto de programas que asseguram a interface com o utilizador, ou entre outros programas e periféricos.
- Ubuntu – uma distribuição específica de linux
 - Conjunto particular de pacotes de software para além do *kernel* (sistema base) que varia ligeiramente conforme o distribuidor.

Outras distribuições de Linux

<i>Distribution</i>	<i>Where to Find It</i>	<i>Description</i>
Slackware	www.slackware.com	One of the original Linux distribution sets, popular with Linux geeks.
Red Hat	www.redhat.com	A commercial business distribution used mainly for Internet servers.
Fedora	www.fedoraproject.org	A spinoff from Red Hat, designed for home use.
Gentoo	www.gentoo.org	A distribution designed for advanced Linux users, containing only Linux source code.
Mandriva	www.mandriva.com	Designed mainly for home use (previously called Mandrake).
openSuSe	www.opensuse.org	Different distributions for business and home use (now owned by Novell).
Debian	www.debian.org	Popular with Linux experts and commercial Linux product developers.

Linux – os componentes

- O *Kernel*

- Inicia e gere programas e permite o acesso a recursos do computador (*hardware*)

- Projecto GNU – open source

- Proporciona comandos, bibliotecas, *shells*, ...

- Xfree86

- Sistema livre que implementa o X Window

Comparando Linux com Windows

Licensing & costs

Linux is an open source OS whereas Windows is a proprietary commercial OS.

Hardware compatibility

A modern Linux may run better than a recent version of Windows on old hardware. Linux also tends to be less resource-intensive, so you can be productive on older hardware when using Linux.

Software availability

Some popular desktop applications, such as Microsoft Office, are available on Windows but not on Linux. Popular server programs, such as the Apache Web server, were developed first for Linux.

User interfaces

Like Mac OS X, Windows uses its own unique user interface. This fact contributes to poor inter-OS portability.

Configurability

Linux is a much more configurable OS than is Windows.

Security

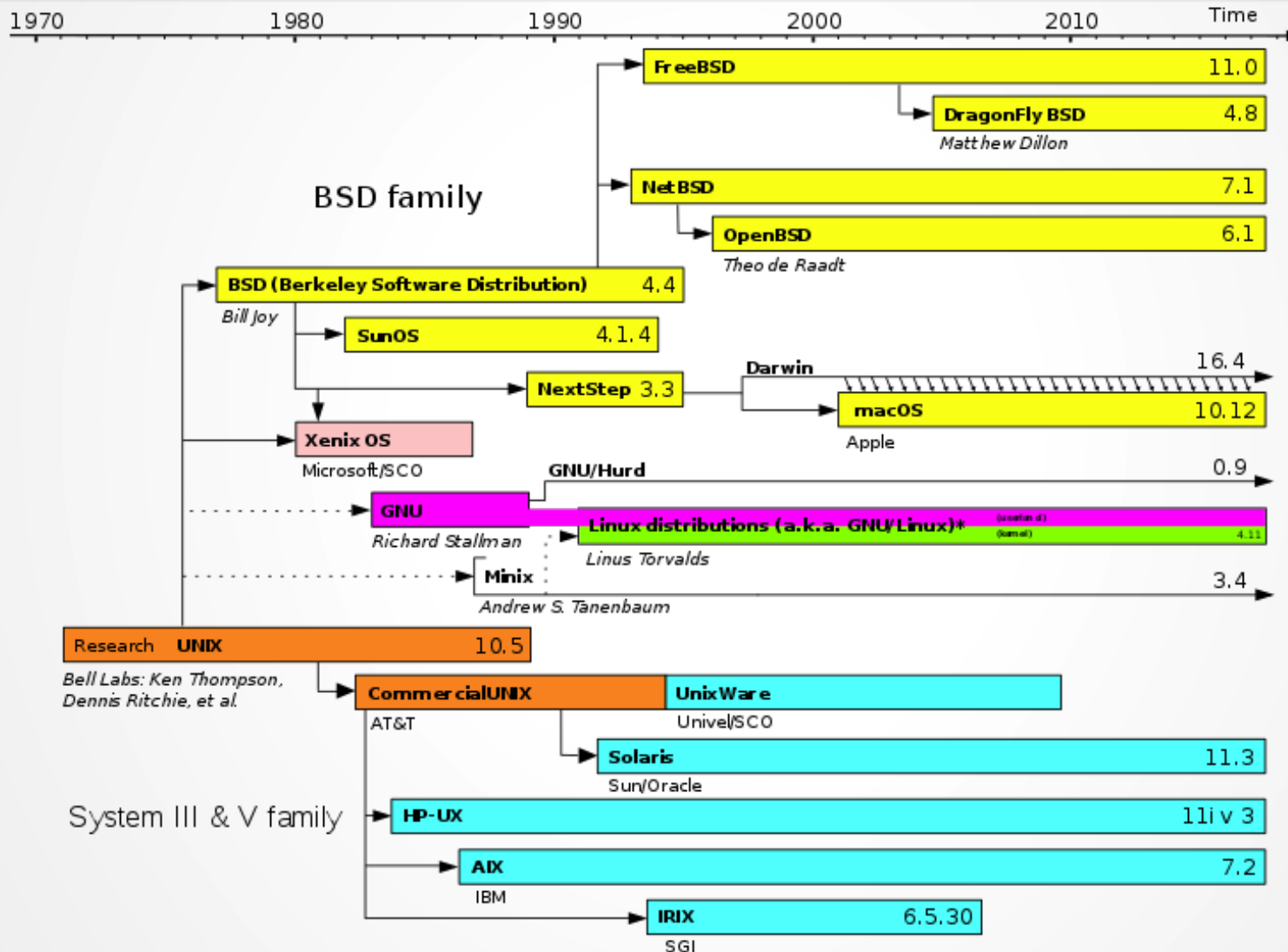
Many of the threats to Windows come from viruses.

Viruses are essentially a non-issue for Linux; in Linux, security threats come mostly from break-ins involving misconfigured servers or untrustworthy local users.

LINUX é inspirado no UNIX

- Sistemas operativos com grandes similaridades:
 - Sistema de ficheiros poderoso e versátil
 - Política de segurança e permissões robusta
 - Os processos (são instâncias de programas)
 - À exceção do processo inicial (`init` – PID 1), todos os processos são “filhos” de outros processos.
 - Sistema *Multi-task* (multi-tarefa)
 - Sistema Multi-utilizador
 - Interação com o utilizador
 - Interpretador de comandos – A *shell*
 - Interface gráfica (X Window)
 - Várias janelas com aplicações específicas (incluindo outras *shells*!)

Do UNIX as LINUX



*The penetration of GNU utilities varies between distributions, some projects use GNU's implementation of the Linux kernel (Linux-libre). Some operating systems mentioned here include GNU utilities to a lesser degree.

SOs concorrentes...?



Imagem originalmente em: <http://www.tauxeef.com/why-linux-is-better-than-windows-linux-rules/>

Interface com utilizador

- Clássico – “*terminal-oriented*”
 - Um único ecrã sem acesso a rato
 - Interface baseada em caracteres ASCII
 - Ubuntu arranca com 6 desses terminais:
 - CTRL+ALT F1, F2, ..., F6
- X Window – sistema de janelas
 - Forma similar ao Windows da Microsoft, ...
 - Aplicações gráficas
 - Ubuntu arranca com uma consola gráfica:
 - CTRL+ALT F7 (ou CTRL+ALT F8)

Porque usar linha de comando?

In our modern age, the GUI is mistakenly considered “progress.” For instance, users of the Microsoft and Apple-based operating systems are quite accustomed to using a mouse to navigate and perform various tasks. While it’s handy in certain situations—it would be difficult to imagine image editing without a mouse, for example—in many other situations, such as when manipulating files, directly typing commands is far more efficient.

In Beginning ubuntu linux, 2008 - pp. 305.

Vantagens da linha de comandos

- Simples e rápido
- Versátil
- Consistente nas diversas distribuições
- Crucial na resolução de problemas
- Útil para acesso remoto

Procurando Ajuda

Comandos úteis em Linux

- **man** (manual de um comando)
- **info** (é uma alternativa similar)
- **help** (ajuda de comandos intrínseco da bash)
- **apropos** (lista comandos que referem um dado assunto)
- **whatis** (descrição resumida de um comando)
- **whereis** (dá a localização de um comando)
- **locate** (localiza caminho ficheiros e pastas)

Exemplo de “whatis”

- Na linha de comando escrever

```
whatis ls
```

- A resposta indica o que é ls e diz em que secção do manual está descrita (1)

```
- ls (1)          - list directory contents
```

- Experimentar sobre outros comandos:

```
sudo, rm, kate, find, mv, locate
```

- Incluindo o próprio `whatis` :-)

- Se o comando for desconhecido a resposta será:

```
Nothing appropriate
```

Ajuda mais completa

- O comando `man` fornece informação total sobre um qualquer outro comando. Por exemplo, o comando

```
man ls
```

- Abre uma página detalhada sobre o comando `ls` no terminal.

- Existem variantes gráficas como o `gman` ou o `xman`
 - O `gman` pode precisar de `man2html` para ver num browser
 - Se porventura o `xman` não estiver instalado, pode-se instalar diretamente da internet com o comando:

```
sudo apt install x11-apps
```

A estrutura de ficheiros

- Ficheiros e diretórios (*directory*)
- A árvore de diretórios que começa em /
- Caminho (*path*) de um ficheiro
- A barra / como separador de diretórios
 - (nota: é diferente do Windows)
 - A barra usada no MS Windows “\” tem um significado especial em Linux, mormente quando se utiliza em parâmetros na linha de comando

Alguns diretórios base mais relevantes

/bin	Essential commands that everyone needs to use at any time.
/boot	The information that boots the machine, including your kernel.
/dev	The device drivers for all the hardware that your system needs to interface with.
/etc	The configuration files for your system.
/home	The home directories for each of your users.
/lib	The libraries, or the code that many programs (and the kernel) use.
/media	A spot where you add temporary media, such as floppy disks and CD-ROMs;
/mnt	A spot where you add extra filesystem components such as networked drives and items you aren't permanently adding to your filesystem but that aren't as temporary as CD-ROMs and floppies.
/opt	The location that some people decide to use (and some programs want to use) for installing new software packages, such as word processors and office suites.
/proc	Current settings for your kernel (operating system).
/root	The superuser's (root user's) home directory.
/sbin	The commands the system Administrator needs access to.
/srv	Data for your system's services (the programs that run in the background).
/sys	Kernel information about your hardware.
/tmp	The place where everyone and everything stores temporary files.
/usr	A complex hierarchy of additional programs and files.
/var	The data that changes frequently, such as log files and your mail.

Exercício

- Num terminal (consola) escrever sucessivamente os comandos:
 - `cd /`
 - `tree -d -L 1 -C | more`
- O resultado é a visualização em forma de árvore do primeiro nível de diretórios a partir da raiz (/)
 - NB: Pode-se dar o caso do comando `tree` não estar instalado: Nesse caso deve fazer-se:
 - `sudo apt install tree`
- Como fazer para visualizar até dois níveis?
- Pode-se saber mais sobre o comando `tree` consultando o seu manual com o comando:
 - `man tree`

Utilizadores e permissões

- Há vários tipos de utilizadores mas há dois principais:
 - Utilizadores comuns (pessoas)
 - Superutilizador (*superuser* ou *root user*)
 - Utilizador com permissões especiais, nomeadamente a permissão de mudar todas as permissões!
 - Qualquer utilizador se pode tornar (temporariamente) num *superuser* mediante o uso de uma *password*
 - Comando “su” ou “sudo”

Permissões de ficheiros

- As permissões sobre ficheiros e diretórios são de três tipos:
 - read (r) , write (w), execute (x)
 - Ou seja , leitura, escrita e execução
- Estas permissões são categorizadas por:
 - user (u), group (g) , other (o)
 - Ou seja, aplicam-se separadamente ao utilizador (dono do ficheiro ou diretório), aos utilizadores do grupo associado e finalmente aos outros (restantes utilizadores)

Significado das permissões

- Quando se faz uma listagem de um ficheiro, por exemplo com:

- `ls -l /bin/dir`

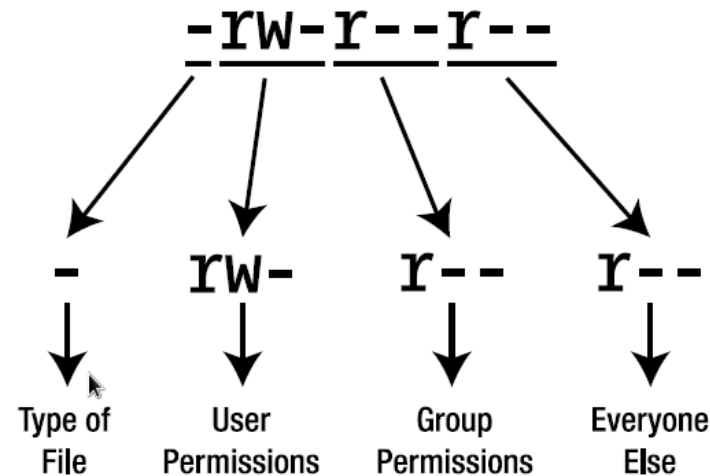
- Surge algo como

```
-rwxr-xr-x 1 root root 133792 jan 18 2018 /bin/dir
```

- O significado é:

-	r	w	x	r	-	x	r	-	x
tipo	Permissões do utilizador proprietário (u)			Permissões de utilizadores do mesmo grupo (g)			Permissões dos restantes utilizadores (o)		

Tipos de ficheiros



- **Code** **File Type**
- - Standard file
- d Standard directory
- l Symbolic link (a shortcut to another file)
- p Named pipe (a file that acts as a conduit for data between two programs)
- s Socket (a file designed to send and receive data over a network)
- c Character device (a hardware device driver, usually found in /dev)
- b Block device (a hardware device driver, usually found in /dev)

Ainda as permissões -2

- É possível mudar o dono (*user*) de um ficheiro bem como o grupo.
- Os comandos são respetivamente
 - `chmod`
 - `chgrp`
- Porém, isso só é em geral possível com permissões de *superuser* (root) e nesse caso o comando tem de ser precedido do comando adicional “sudo”:
 - `sudo <comando-de-utilizador>`

Exercício

- Criar um ficheiro com a listagem do diretório *home* do utilizador
 - `ls -l ~ > ~/list`
- Mudar-lhe as permissões para não ser legível ao dono apenas
 - `cd`
 - `chmod u-r list`

Tentar ver o conteúdo com o comando `cat`

- Perante o erro, repor as permissões (`u+r`) e voltar a tentar ver o conteúdo.

Edição de ficheiros

- Interativa (no terminal ou com aplicações dedicadas a esse efeito)
 - `vi`, `gedit`, `kate`, `emacs`, `textedit`, `kwrite`, `code` (antigo `vscode`), ...
- Programada (linha de comandos)
 - `awk`, `sed`, `ed`, ...
 - Vantajosa para:
 - edição rápida de grandes ficheiros
 - edição de fluxos contínuos de informação (“ficheiros” sem limites)

As primeiras aplicações X – um terminal

- Criar um terminal X (`gnome-terminal`)
 - Antigamente `xterm` era o nome usado e embora ainda exista já se usa pouco.
- Os parâmetros da linha de comandos
 - Colocar o terminal numa dada região do ecrã e com dadas dimensões (80 colunas, 40 linhas):
`gnome-terminal --geometry 80x40+200+30`
 - Para usar outras opções:
 - Ver manual (para os dois comandos)
 - `man gnome-terminal`
 - `man xterm`

A “SHELL”

- Programa que corre num terminal (*caractere ou gráfico*)
- Aceita comandos do utilizador
- Lança aplicações (*gráficas ou não gráficas*)
- Executa ficheiros com sequências de comandos

Tipos de SHELL

- Bourne Shell (*sh*)
 - Das mais antigas
- Bourne Again-Shell (*bash*)
 - É a mais comum atualmente!
- Outras
 - C-Shell (*csh*)
 - Korn-shell (*ksh*)
 - Z shel (*zsh*)
 - ...

Alguns comandos de SHELL

- `ls`
- Os caracteres especiais da shell
 - `*` `.` `?` `{ }`
- `echo`
- `mkdir`
- `rm`
- `mv`
- `cp`

`cat`

`ps`

`grep`

`find`

Uso de variáveis

`$`

...

Redirecionar output e input

- Por defeito muitos comandos de *shell* e outros têm como *output* o ecrã e como *input* o teclado.
- O *output* pode ser redirecionado para um ficheiro (ou para “pipes”,...)
- O *input* pode ser obtido de ficheiros (ou de “pipes”, ...)
- Exemplo de *output*:

```
echo "abc"
```

```
echo "abc" > file1
```

```
echo "abc" >> file1
```

- No 1º caso o texto “abc” é escrito no ecrã (*default*)
- No 2º caso texto “abc” é escrito num ficheiro designado por “file1”
- No 3º caso texto “abc” é “acrescentado” ao ficheiro “file1” (se não existir é criado)

Acesso ao hardware

- Os dispositivos de hardware são ficheiros especiais para onde se “escreve” e de onde se “lê”
- Exemplo: uma porta série é um “ficheiro” com um nome similar a `/dev/ttyS0`.
 - Escrever para a linha série faz-se com editores não interativos

Os “PIPES”

- Conceito – o *output* de um comando é o *input* de outro.
- Exemplo:
 - listar todos os processos de sistema que tenham a expressão “getty”:

```
ps -e | grep getty
```

NB. As opções de certos comandos podem variar de implementação para implementação (System V, BSD, Berkeley,)

Exemplo mais complexo com PIPES

- Contar quantas vezes o computador já fez “*reboot*” conforme o seu registo interno (em geral re-iniciado a cada mês).
- Encadear os comandos:
 - `last` (lista de registos de acesso ao sistema)
 - `grep` (filtragem das linhas com um certo texto)
 - `wc` (conta palavras, letras ou linhas de um texto)

```
last | grep boot | wc -l
```

Exercício com PIPES

- Contar quantas vezes surge nas mensagens de *boot* (comando `dmesg`) uma referência ao bus `pci`.
 - Notar que a referência a `pci` pode surgir em maiúsculas ou minúsculas e por isso o comando `grep` deve ter uma opção adicional (ver manual)
 - Os comandos a encadear são:
 - `dmesg`
 - `grep`
 - `wc`
 - Nota: faltam as opções de `grep` e `wc`

SED e AWK: editores de linha

- Visualizar o ficheiro da lista de utilizadores:
`cat /etc/passwd`
- Com “sed” substituir todos os separadores “:” por “,” no ficheiro /etc/passwd e colocar o resultado num outro ficheiro:
`sed 's/:/,/g' /etc/passwd > temp`
- Com “awk”, em cada linha, obter apenas o 1º e o 4º campo do mesmo ficheiro:
`cat /etc/passwd | awk -F: '{print $1, $4}'`

Um desafio em bash

Executar o comando na bash:

```
for i in {1..50}; do xclock & sleep 0.1 ; done
```

O resultado deve ser o display de 50 relógios distribuídos no ecran!

Como fazer para os fechar todos? Um a um? Moroso...

Pela linha de comando pode-se fazer o “kill” dos processos associados; um de cada vez, ou criar um comando composto para os detetar todos e fazer o kill respetivo. Comandos sugeridos:

```
ps, grep, awk, xargs, kill
```

Como compor estes comandos usando pipes?

```
ps | grep xclock | awk '{print $1}' | xargs kill
```

Exercícios e tutorial em bash

- Para praticar bash recomenda-se fazer o tutorial presente em:
<http://mywiki.woledge.org/BashGuide>
- Pelo menos os capítulos assinalados:
 - Commands and Arguments *
 - Special Characters *
 - Parameters
 - Patterns *
 - Tests and Conditionals
 - Arrays
 - Input and Output *
 - Compound Commands *
 - Sourcing
 - Job Control
 - Practices