

Administração de Sistemas

2018-2019

Linux Parte 1 - Introdução

***Linux/Unix (Baseado na
distribuição CentOS 6.x e 7.x)***

Docente: Armando Ventura



Linux - Introdução

Baseado na distribuição

CentOS 6.X e 7.x

Linux

- Linux uma primeira abordagem
- Distribuições Linux
- O núcleo do sistema Linux
- Estrutura de directórios
- Partições
- Devices
- Aceder a Devices
- O sistema de ficheiros Linux
- Directórios e caminhos
- Protecção de ficheiros e directorias
- Ficheiros especiais – symbolic links

Linux uma primeira abordagem

- O Linux é um sistema operativo, concorrente de outros sistemas como o Windows e o Mac OS, que foi concebido originalmente por um estudante da Universidade de Helsínquia na Finlândia, Linus Torvalds.
- Corre sobre uma arquitectura de 32 ou 64 bits.
- Memória protegida - evita que um erro gerado numa aplicação faça parar todo o sistema, para além de garantir a separação entre os processos gerados pelas aplicações e os processos gerados pelo próprio sistema (KERNEL)
- Permite a sua utilização por vários utilizadores
- Pode integrar uma rede estruturada com TCP/IP
- Multitarefa e Multiutilizador
- Flexibilidade, Economia e Fiabilidade

Distribuições Linux

- Red Hat Enterprise – CentOS
 - Fedora X X
- SuSE
- Caldera
- Mandrake
- Turbo Linux
- Slackware
- Debian
- ...

O núcleo do sistema Linux

- O núcleo de qualquer sistema operativo designa-se por KERNEL. Sem entrar em pormenores, podemos dizer que o KERNEL é a parte do sistema operativo que indica ao processador para fazer o que quer que o utilizador comande.
- O KERNEL não é mais do que um conjunto de componentes e rotinas básicas do sistema operativo que lhe permitem executar programas e outras tarefas, por forma a tornar-se útil ao utilizador.

Estrutura de diretórios

- Partindo da directoria principal “/” que podemos comparar com o “c:\>” dos sistemas windows, onde esta é a directoria raiz do sistema, existe habitualmente um conjunto de directorias que recebem ficheiros fundamentais para o funcionamento do Linux. A lista seguinte pretende resumir o conteúdo de cada uma das principais directorias de um sistema Linux.

Estrutura de Diretórios

- **/bin** Conjunto de programas utilizados durante o processo de arranque.
- **/boot** Programa de arranque básico do sistema. Contém um conjunto de ficheiros que inclui o kernel (núcleo do sistema) do Linux.
- **/dev** Conjunto de ficheiros que representam os dispositivos físicos do sistema.
- **/etc** Conjunto de ficheiros de configuração do sistema operativo (não todos, mas a maioria).
- **/home** Directorias de trabalho dos utilizadores.
- **/lib** Conjunto de bibliotecas necessárias ao funcionamento do sistema e respectivos programas. Dentro desta directoria poderão existir outras entre as quais a **/lib/modules** que contém um conjunto de módulos de software e de drivers que podem ser incluídos no kernel do sistema.
- **/mnt** Designação normalmente utilizada para mostrar o conteúdo dos dispositivos físicos como discos, disquetes, etc.
- **/proc** Contém ficheiros virtuais que representam o estado dos processos em execução e informação sobre os componentes do sistema operativo.
- **/root** Directoria de trabalho do superutilizador (root).
- **/sbin** Contém os principais programas utilizados para administrar o sistema operativo.
- **/tmp** Directoria de ficheiros temporários. Se criar algum ficheiro nesta directoria ele poderá ser eliminado quando o sistema recomeça.
- **/usr** Contém um conjunto de directorias com programas, bibliotecas, documentação, etc.
- **/var** Contém um conjunto de dados do sistema operativo como correio electrónico, ficheiros em fila de impressão, ficheiros que impedem a utilização do mesmo dispositivo em simultâneo (*locks*) por vários utilizadores e ficheiros que registam a actividade do sistema (*logs*).
- **/lost+found** Sempre que não é efectuado o *logout* ao sistema de forma correcta, ou seja quando o computador é desligado com o sistema em funcionamento, podem perder-se ficheiros. Quando o sistema é reiniciado, o disco é verificado e os ficheiros perdidos são enviados para esta directoria. Geralmente existe uma directoria **/lost+found** por cada partição de disco do sistema.

Estrutura de Diretórios (Continuação...)

A directoria **/usr** contém igualmente um conjunto de subdirectorias relevantes:

- **/usr/bin** Contém a maioria dos programas utilitários do sistema.
- **/usr/dict** Dicionários para os correctores de texto.
- **/usr/doc** Diversa documentação sobre o sistema e os programas instalados.
- **/usr/include** Ficheiros com dados e definições do sistema operativo e livrarias utilizadas nas linguagens de programação C e C++.
- **/usr/info** Documentação que inclui alguns manuais sobre Linux.
- **/usr/lib** Ficheiros de bibliotecas dinâmicas que são utilizáveis pelos programas em execução.
- **/usr/man** Directoria onde são guardados os manuais do Linux, normalmente designados por *man-pages* e que podem ser chamados através do comando *man*.
- **/usr/sbin** Directoria onde ficam os principais programas de administração do sistema e os *daemons* que não são mais do que programas que representam os diversos serviços do sistema operativo.
- **/usr/share** Ficheiros de dados partilhados pelos programas e aplicações. Esta directoria pode conter directorias para cada programa ou conjunto de programas.
- **/usr/src** Local onde se encontra o código-fonte do sistema operativo.
- **/usr/X11** O sistema de janelas X Windows e os seus programas.
- **/usr/local** Directorias com pacotes de software opcional que pode ser instalado depois da instalação standard do sistema.

Estrutura de diretórios

A directoria **/var** contém diversas subdirectorias entre as quais podemos destacar:

- **/var/spool** Área onde são guardadas as filas de impressão e correio electrónico.
- **/var/lock** Área onde é registada a utilização de vários serviços e periféricos, para impedir que vários utilizadores e serviços interfiram uns com os outros.
- **/var/run** Ficheiros com identificação dos processos responsáveis pelo funcionamento dos diversos serviços, permitindo esses ficheiros parar ou terminar determinados serviços.
- **/var/log** Ficheiros de registo do eventos do sistema.

Partições

- É possível subdividir um disco rígido em partes, que na prática se comportam como discos independentes, apesar de fisicamente serem um único dispositivo. A estas partes damos o nome de partições, embora em Windows elas possam também ser conhecidas por discos lógicos (*logical disks*).
- Um disco rígido pode ser dividido (“particionado”) diversas vezes, podendo mesmo receber diferentes sistemas operativos em cada partição. Cada partição de um disco pode ser formatada independentemente e geralmente é efectuada durante o processo de instalação do sistema operativo.
- Se o disco rígido de determinada máquina foi particionado, cada partição é considerada como um dispositivo independente. Por exemplo, se um disco rígido é representado por a drive **/dev/hda**, então a primeira partição deste disco deverá ser representada por **/dev/hda1**, a segunda por **/dev/hda2** e assim sucessivamente.



Disco do 1º controlador IDE

Devices

Nos sistemas Linux/UNIX, a palavra *device* é utilizada para designar todos os dispositivos periféricos ligados ao computador – o

que inclui discos rígidos, drives de disquetes, drives de CD-ROM, placas de som e vídeo, portas de série, portas paralelas e

todos os outros dispositivos. A cada *device* é dado um nome – exemplo **/dev/hda**.

- Os nomes mais comuns utilizados para representar dispositivos em Linux/UNIX são:
- **/dev/hd*** onde * pode ser a,b,c,... estes dispositivos podem ser drives de disquetes, disco rígido, cd-rom, etc, do tipo IDE.
- **/dev/hda** – representa o dispositivo MASTER ligado ao primeiro controlador IDE (normalmente será a primeira drive de disco, que no Windows / DOS seria representada por C:).
- **/dev/hdb** - é a drive ligada como SLAVE no primeiro controlador IDE (que poderá ser um segundo disco rígido ou uma drive de CD-ROM por exemplo. E assim por diante dependendo do número de dispositivos instalados.
- **/dev/sd*** onde * pode ser a,b,c,... estes são dispositivos SCSI, por exemplo discos rígidos.
- **/dev/fd*** onde * pode ser 0,1,...são drives de disquetes. /dev/fd0 é a primeira drive de disquetes que seria representada no Windows com drive a:, /dev/fd1 é a segunda drive de disquetes, b: no Windows e assim sucessivamente.
- **/dev/lp*** onde * pode ser 0,1,... são portas paralelas. As portas paralelas, são em geral, utilizadas para ligar impressoras ao computador. Em Linux o conteúdo da primeira porta paralela, LPT1 no Windows, seria representado debaixo da directoria /dev/lp0; dev/lp1 seria a porta LPT2, e assim sucessivamente.
- **/dev/ttyS*** onde * pode ser 0,1,... são portas serie. Estas portas são geralmente utilizadas para ligar dispositivos como o rato ou um modem. A directoria /dev/ttyS0 corresponde à porta COM1 no Windows, /dev/ttyS1 à porta COM2 e assim sucessivamente.
- **/dev/audio** e **/dev/dsp** são dois nomes utilizados para representar a placa de som. São dois dispositivos diferentes, visto serem utilizados para diferentes tipos de ficheiros de som.
- Para além destes *devices* é comum a existência de *symlinks* (aqueles que são apenas apontadores) para cada um dos dispositivos **/dev/floppy** (drive de disquetes), **/dev/modem** e **/dev/cdrom**.

Aceder a Dispositivos

- O sistema tem de receber uma instrução para incorporar o conteúdo da disquete ou CD na árvore que representa todas as directorias do sistema. A esta acção dá-se o nome de *mount* – podemos dizer que montámos o dispositivo de CD, por exemplo.
- Tipicamente o conteúdo do CD aparecerá na árvore de ficheiros debaixo do nome **/mnt/cdrom**; a drive de disquetes debaixo do nome **/mnt/floppy** aos quais se dá a designação de **mount points** e podem ser configurados num ficheiro especial **/etc/fstab**.
- Aceder a uma drive deste modo, não significa que o sistema vai copiar o conteúdo do CD ou da Drive de disquetes para uma directoria **/mnt/cdrom** ou **/mnt/floppy**, apenas quando um programa tenta aceder a um ficheiro no CD, por exemplo **/mnt/cdrom/index.html**, é que o sistema irá à procura do ficheiro na drive de CD.
- Comandos a utilizar:
 - mount (para montar device)
 - umount(para desmontar device)

O sistema de ficheiros Linux

- Uma referência importante vai para o facto de o Linux ser “case- sensitive” o que significa que um ficheiro com o nome **fichl.txt** é diferente do ficheiro com o nome **Fichl.txt**, podendo portanto coexistir. Para além desta regra existem uma série de convenções sobre ficheiros, herdadas do sistema UNIX, que embora não sejam obrigatórias devem ser observadas.
 - É habitual utilizar o formato **nomeficheiro.extensão** para os ficheiros, em que a extensão designa o tipo do ficheiro. Por exemplo, para os ficheiros de texto simples é utilizada a extensão **.txt** .
 - As directorias começadas por ponto (.), são de um modo geral ficheiros de configuração, isto é, são directorias que guardam ficheiros de configuração de determinada aplicação ou do sistema.
 - Os ficheiros cujos nomes completos terminam em ~ são, de um modo geral, ficheiros temporários criados por aplicações. Por exemplo, ao editar o ficheiro **texto.txt** é guardado o ficheiro temporário **texto.txt~** .

O sistema de ficheiros Linux

wildcards

- Quando utiliza o Linux em modo de linha de comandos, pode utilizar aquilo que normalmente é designado por *wildcards* em vez de um nome completo do ficheiro.
- À semelhança daquilo que acontece no DOS, o Linux utiliza o `*` como caracter de substituição de um conjunto de caracteres indefinido. Sem querer entrar em detalhes sobre os comandos Linux, podemos mostrar como exemplo os comandos:
 - **ls *.txt** que mostra a lista de todos os ficheiros com extensão txt
 - **rm capitulo*** que elimina todos os ficheiros cujos nomes comecem por **capitulo**.
- Outro caracter de substituição bastante utilizado é o `?` que pode substituir qualquer símbolo isolado dentro de uma cadeia de caracteres. Por exemplo:
 - **rm capitulo?.txt** eliminaria os ficheiros capitulo1.txt e capitulo2.txt mas não o ficheiro **capitulo10.txt**
- Espaços e caracteres especiais nos nomes dos ficheiros
 - O nome de um ficheiro pode conter letras, números, espaços, vírgulas, todos os caracteres excepto (`/`). No entanto, se está a utilizar a linha de comandos, é necessário algum cuidado com estes caracteres, sobretudo os espaços. Assim, é aconselhável a utilização de nomes entre pelicas ("`"`) sempre que tiver necessidade de se referir a ficheiros que contenham espaços no nome. Por exemplo:
rm 'O Meu Ficheiro' em vez de **rm O Meu Ficheiro**

Directórios e caminhos

- O caminho de um ficheiro pode designar-se por **nome completo do ficheiro**, visto conter não só o nome do ficheiro mas também a sua localização dentro da estrutura de directórios da unidade onde é gravado, podendo um directório localizar-se dentro de outro. Deste modo, um caminho para um ficheiro poderia representar-se do seguinte modo:

`/home/vsf/teste.txt`

- O nome do ficheiro é **teste.txt** que está contido no directório **vsf** que por sua vez está contido no directório **/home**.

O directório e o nome do ficheiro estão separados por `/` e por esta razão os nomes dos ficheiros não podem conter o carácter `/`. Esta estrutura é semelhante à utilizada no DOS e no Windows só que com a barra contrária `/` e não `\`. O directório que contém outro é designado por *parent directory* desse directório. No exemplo anterior, o directório **/home** é o *parent directory* do directório **vsf**.

- Caminho absoluto e caminho relativo
- Convenções utilizadas para caminhos de ficheiros
 - `~/` directoria **home** do utilizador
 - `./` directoria de trabalho actual (corrente)
 - `../` directoria anterior (parent) à directoria corrente

Protecção de ficheiros e directorias

- Num sistema multi-utilizador é necessária a existência de um mecanismo que proteja os dados e programas de cada utilizador face aos outros utilizadores.
- Assim, no Linux existe um sistema que permite definir aquilo que pode ou não ser feito sobre um ficheiro e quais utilizadores que o podem fazer, através do que designamos por **sistema de permissões**.
- Este sistema de protecção de ficheiros e directorias é feita através de um conjunto de nove *marcas*, conhecidas por **bits ou atributos de protecção**, agrupadas em 3 classes, tendo assim cada classe ou subdirectoria 3 bits, conforme seguidamente discriminadas pela sua ordem:
 - 1. **Nós próprios** – o utilizador, dono do ficheiro ou directório.
 - 2. **O grupo** – os utilizadores pertencentes ao mesmo grupo do nosso.
 - 3. **Todos** – todos os utilizadores do sistema.
 - Os bits de protecção são referidos pelas seguintes letras:
 - **r** Autorização para leitura (*read*) de um ficheiro (ou directoria, ou seja aceder aos dados para consulta).
 - **w** Autorização de alterar, apagar e renomear o ficheiro ou directoria (*write*). No caso das directorias que pode incluir e excluir ficheiros dessas mesmas directorias.
 - **x** Autorização de execução (*execute* ou *search bit*), que permite a utilização do ficheiro como programa ou como um *script* para a consola. No caso das directorias, permite a utilização do nome da directoria para aceder ao seu conteúdo, mas não permite vê-lo. Apenas através da combinação dos direitos **read(r)** e **execute(x)** permite listar o conteúdo de uma directoria.

Protecção de ficheiros e directorias

- Os três atributos de cada classe podem ser combinados a fim de se obter um determinado nível de segurança. Por via de regra, a presença de uma letra como atributo (ou marca) de protecção numa classe, representa a autorização representada pelo significado da letra (**r**) para leitura, **w** para escrita e **x** para execução). Por sua vez, a sua ausência (sendo então) utilizado o caracter **-** (menos) representa a negação desse atributo, conforme mencionado abaixo:
 - **rwxr--r--** O dono do ficheiro (ou directoria) pode ler, escrever e executar. Todavia, o grupo e todos os restantes utilizadores apenas pode lê-lo.
 - **rwxrwxrwx** Dono, grupo e mundo (todos os utilizadores do sistema) podem ler, editar e executar.

Protecção de ficheiros e directorias

- A alteração dos bits de segurança pode ser realizada através de várias técnicas. Uma técnica pode ser através do comando **chmod** com o auxílio da linguagem binária, onde o bit **1** simboliza a presença de um determinado atributo e o bit **0** a sua ausência. A combinação binária obtida (9 bits) deve ser convertida num número octal, conforme seguidamente exemplificado:
 - **rwxr--r--** 111100100 744 (em octal)
 - **r-xr-xr--** 101101100 554
- Assim sendo, supondo que se pretenda o ficheiro **pl.c**, com as seguintes propriedades:
 - **rwx--x--x** 111001001 711 (em octal) então, o comando a emitir na linha de comando deverá ser **chmod 711 pl.c**
 - **chmod a-wx ***
all (utilizador, grupo e outros) têm acesso de leitura (**read**) e não de escrita (**write**) e de execução a todos os ficheiros.
 - **chmod u+rwx g=rx o-wx .**
o utilizador pode ler, escrever e executar a directoria corrente, o grupo tem somente permissão de leitura e execução. Aos outros, é-lhes retirada a capacidade de escrita e de execução.

Ficheiros especiais – symbolic links

- Para além dos ficheiros normais, existe no Linux um conjunto de ficheiros especiais aos quais se dá o nome de **symbolic links** ou **symlinks**. Estes ficheiros não contêm quaisquer dados de informação, em vez disso, são apenas apontadores ou atalhos para outros ficheiros (tal como acontece no Windows com os shortcuts). Por exemplo, o nosso utilizador 'luisa' pode ter um symlink **istec.txt** que aponta para o ficheiro **docs/cadeira1.txt**. Desta forma, sempre que um programa tentar aceder ao ficheiro **istec.txt** o ficheiro que será aberto será o ficheiro **docs/cadeira1.txt**. Como pode ver por este exemplo, o ficheiro e o symlink podem ter nomes diferentes e localizações distintas.
- Note que, apagar, mover ou renomear o symlink não tem qualquer efeito no ficheiro para o qual ele aponta. Assim, se a Luísa apagar o ficheiro **istec.txt**, é apenas o symlink que é apagado, o ficheiro **docs/cadeira1.txt** não será afectado. Do mesmo modo, as permissões que podem ter sido atribuídas ao symlink ou à directoria em que ele está contido não têm qualquer significado, são as permissões do ficheiro propriamente dito que determinam as permissões dos utilizadores face a esse ficheiro.
- Os symlinks podem referir-se não apenas a ficheiros mas a directorias.

To make a symbolic link use the command:

ln -s source linkname

Exemplos:

- To make a symbolic link from a file in one subdirectory to a file in another subdirectory:
ln -s reports/reportA publications/my_report This makes a symbolic link between the file reportA in the subdirectory reports and the filename my_report in the publications subdirectory.
- To make a symbolic link to a file in a subdirectory to your current working directory:
ln -s /docs/editors /vi.ref This makes a symbolic link between the file vi.ref in the subdirectory docs/editors/ to the filename vi.ref in the current working directory.