

Introdução à Utilização de Sistemas Unix

Versão 1.0

Hugo Miranda



Agosto de 1999

Introdução

Os objectivos deste texto são:

1. Servir como material didáctico de apoio à cadeira de Fundamentos de Sistemas Operativos das licenciaturas do Departamento de Informática
2. Introduzir os alunos na utilização dos recursos informáticos do departamento

O primeiro através da apresentação do sistema operativo Unix e da sua versão para computadores pessoais: o Linux que serão objecto de estudo no decorrer do semestre.

O segundo, e uma vez que a cadeira está inserida no primeiro ano das licenciaturas, através da apresentação das operações básicas de utilização dos sistemas operativos das máquinas instaladas nos laboratórios¹. Pretende-se assim diminuir o esforço de aprendizagem dos novos alunos.

A informação aqui apresentada é apenas a iniciação ao mundo destes dois sistemas operativos. Para além das funcionalidades próprias, o Linux tem crescido graças à disponibilização gratuita de inúmeros programas pelos seus autores. A descoberta do mundo Linux faz também, embora informalmente, parte do curso.

Este documento está dividido em 4 partes. Na primeira são introduzidos os conhecimentos elementares que os utilizadores deverão ter para uma utilização correcta do equipamento do departamento bem como as responsabilidades que a sua utilização acarreta. Na segunda são apresentados os sistemas Unix e Linux, os seus comandos e enumeradas as suas diferenças. A terceira parte introduz o ambiente gráfico X Windows para estes sistemas operativos e que também está disponível nos laboratórios. Finalmente, a quarta parte apresenta um conjunto de ferramentas que, não pertencendo ao sistema operativo, o complementam oferecendo facilidades adicionais.

Este trabalho é uma extensão e adaptação do documento “*Introdução à Utilização do Sistema Unix*” publicado pelo Departamento de Informática em 1990 [PDMM90].

Convenções

No texto que se segue serão utilizadas algumas convenções que a seguir se descrevem:

1. Os exemplos de diálogos no terminal são representados por uma fonte tipográfica diferente.
2. Alguns comandos são introduzidos pela combinação de mais de uma tecla. As teclas Ctrl (Control) e Alt(Alternate) são utilizadas da mesma forma que a tecla Shift: mantêm-se premidas durante a digitação de uma outra tecla. Ao longo deste documento utiliza-se o símbolo $\hat{}$ para representar a tecla Ctrl. Assim, \hat{C} significa premir e manter premida a tecla Ctrl, premir C e em seguida libertar Ctrl. Combinações com a tecla Alt

¹Um deles o Linux.

serão apresentadas na forma Alt+Tecla. Por exemplo, Alt+F representa a pressão simultânea de Alt e F.

3. As teclas de função situam-se no topo do teclado, identificadas por Fn em que n é um número entre 1 e 12. Se não for referido o contrário as sequências Fn, por exemplo F5, representam as teclas de função e não a sequência de caracteres F e 5.
4. A tecla Enter é a tecla que termina a introdução de um comando ou que, nos editores de texto, provoca a descida de uma linha. Os teclados convencionais dispõem de duas teclas equivalentes: à direita do bloco principal do teclado (também representada por ↵) e à direita do bloco numérico, representada pelo texto Enter. Sempre que necessário, utilizar-se-á a expressão <Enter> para a denominar.
5. A tecla denominada por backspace é a tecla que se encontra, nos teclados convencionais, por cima da tecla Enter e que permite apagar o carácter à esquerda do cursor. Nalguns teclados é representada por ←.
6. A introdução de comandos termina sempre com a tecla <Enter> excepto quando indicado o contrário.
7. Na especificação de comandos Unix serão utilizadas sequências especiais. Assim, [x] denotará x zero ou uma vez e {x} representará x zero ou mais vezes.
8. A tradução de alguns termos para português não é usual e deturpa o seu significado. Esses termos, como login, logout ou shell, não serão traduzidos.

Conteúdo

I	Utilização dos Computadores dos Laboratórios	1
1	Responsabilidades dos utilizadores	3
1.1	Passwords	4
2	O Linux Loader	5
3	O Windows NT	7
3.1	Login e Logout no Windows NT	7
3.2	Persistência dos Dados	8
3.3	Utilização do software	9
3.4	Mudar do sistema operativo Windows NT para Linux	9
II	O Sistema Operativo Unix	11
4	Unix e Linux	13
5	Conceitos elementares	15
5.1	Persistência dos dados	15
5.2	Entrada e saída de uma sessão em Linux	16
5.3	Reinicialização de um sistema Linux	17
5.4	Sequências de controlo	17
5.5	Letras maiúsculas e minúsculas	17
5.6	Administrador de sistema	18
5.7	Ficheiros	18
5.7.1	Nomes dos ficheiros	18
5.7.2	Meta-caracteres	19
5.8	Directorias	20
5.8.1	Directoria corrente	21
5.8.2	A directoria HOME	22
6	Execução de comandos	23
6.1	Os interpretadores de comandos do Unix	23
6.1.1	Variáveis de shells	23
6.2	Formato dos comandos	24

7	Comandos do Unix	27
7.1	Modificação da password	27
7.1.1	Exemplo de utilização	27
7.2	Manual interactivo	28
7.2.1	Comando man	28
7.2.2	Comando apropos	29
7.2.3	Comando whatis	29
7.3	Listagem de ficheiros em directorias	29
7.3.1	Opções mais comuns	30
7.4	Operações sobre directorias	31
7.4.1	Mudança de directoria corrente	31
7.4.2	Determinação da directoria corrente	31
7.4.3	Criação/eliminação de directorias	31
7.5	Visualização do conteúdo de ficheiros	31
7.5.1	Comando cat	32
7.5.2	Comandos more e less	32
7.6	Operações sobre ficheiros	32
7.6.1	Eliminação de ficheiros	32
7.6.2	Cópia de ficheiros	33
7.6.3	Alteração do nome de ficheiros	33
7.6.4	Impressão de ficheiros	33
7.7	Protecção de ficheiros	34
7.7.1	Verificação dos direitos	35
7.7.2	Representação numérica	35
7.7.3	Modificação dos direitos	36
7.8	Informação sobre outros utilizadores do sistema	37
7.8.1	Utilizadores activos	37
7.9	Comandos de processos	38
7.9.1	Execução de comandos em <i>background</i>	38
7.9.2	Matar um processo	38
7.10	Restrições de espaço	39
III	O Sistema de Janelas X Windows	41
8	Apresentação do X Windows	43
9	Entrada e saída de uma sessão de X	45
10	Elementos básicos de uma sessão de X	47
10.1	Écrans virtuais	47
10.2	Rato	48
10.3	Menus	48
10.4	Utilização	49
10.4.1	Shells	49
10.4.2	Invocação de aplicações	49
10.4.3	Abertura de sessões remotas	49
10.4.4	Cópia de dados	50

11 Gestores de janelas	51
11.1 Ficheiros relevantes	51
IV Utilitários	53
12 Editores de texto	55
12.1 Visual Editor	55
12.1.1 Modos de edição	55
12.1.2 Outras funções	56
12.2 Joe	56
12.2.1 Ficheiro de configuração	56
12.3 emacs	57
13 Correio electrónico e newsgroups	59
13.1 Regras de utilização de newsgroups	60
13.2 Pine	60
13.2.1 Écran	61
13.2.2 Configuração	61
14 Login remoto	63
14.1 Secure Shell (ssh)	63
14.1.1 Problemas de autenticação	63
15 Arquivo e compressão	65
15.1 Tape Archive Utility (tar)	65
15.1.1 Exemplos de utilização	66
15.2 GNU Zip	66
15.3 Zip	67
15.4 Compress	67
16 Cópia de Ficheiros	69
16.1 File Transfer Protocol	69
16.1.1 Modo de transferência	70
16.2 Secure Copy (scp)	70
16.2.1 Exemplos de utilização	71
16.3 Acesso a drives de diskettes	71
16.3.1 Exemplos de utilização	72
16.4 Split	73
16.4.1 Exemplos de utilização	73
A Resumo dos comandos do GNU Emacs	75
B Resumo dos comandos do joe	79
C Resumo dos comandos do vi	81

Lista de Figuras

3.1	Janela de encerramento de sessão do sistema operativo Windows NT	8
5.1	Funcionamento de um sistema de ficheiros distribuído	16
5.2	Exemplo de uma árvore de directorias	20
10.1	Écran de dimensões superiores ao monitor	47
10.2	Écran de dimensões superiores ao monitor paginado	48
10.3	Execução de aplicações em máquinas remotas no X Windows . .	50

Lista de Tabelas

5.1	Sequências de controlo do Unix	17
7.1	Protecções de ficheiros e valor octal correspondente no sistema Unix	36
16.1	Resumo dos comandos do programa ftp	70
16.2	Alguns comandos da família <i>mtools</i>	72

Parte I

Utilização dos Computadores dos Laboratórios

Capítulo 1

Responsabilidades dos utilizadores

A disponibilização aos alunos dos equipamentos do Departamento de Informática nos termos em que o é actualmente, representa uma prova de confiança deste nos primeiros. Esta abertura pressupõe que os alunos detêm a responsabilidade e um conjunto de conhecimentos que não ponha em causa o seu correcto funcionamento. Assim, é responsabilidade destes o zelar pela integridade do equipamento (hardware e software).

A cada aluno é atribuída uma identificação pessoal que lhe permite a utilização dos equipamentos e espaço em disco¹ para conservação de ficheiros fora do âmbito das cadeiras. Sempre que necessário, são abertas áreas para utilização no contexto das cadeiras.

O acesso a um meio partilhado pressupõe o cumprimento de algumas regras (muitas delas de bom senso) por forma a não prejudicar os restantes utilizadores e que aqui são vagamente abordadas.

Todos os alunos são responsáveis pelas acções tomadas em seu nome, tenham ou não sido executadas pelos próprios. Chama-se por isso a atenção para a necessidade de manter secretas as suas palavras-passe² e de não executar quaisquer programas que possam por em causa a integridade do sistema.³ Os autores destes programas, normalmente estranhos ao departamento, usam os relacionamentos que estabelecem em *newsgroups* ou *chats* para convencer utilizadores legítimos dos equipamentos a os executarem.

Também a introdução de vírus nos sistemas operativos devido à irresponsabilidade de executar programas de origens suspeitas deve ser tida em atenção. A instalação de software nas máquinas está reservada ao pessoal devidamente autorizado, sendo proibida aos restantes utilizadores.

Finalmente, os utilizadores não devem sobrecarregar as máquinas que utilizam com a execução de programas fora do âmbito do departamento e gerir as áreas que lhes são concedidas de forma sensata.

Por último, a escassez dos recursos oferecidos pelo departamento obriga a que os alunos usem do maior bom senso na sua ocupação, devendo limitar-se ao

¹Vulgarmente designado por área

²A sua cedência a qualquer pessoa é proibida pelas normas do Departamento de Informática

³Estes programas são vulgarmente denominados por *cavalos de Tróia*.

tempo estritamente necessário para a realização dos seus trabalhos.

1.1 Passwords

Sendo a password o comprovativo, aos olhos do sistema, de que se é quem se diz ser, torna-se importante evitar que terceiros se apoderem dela, o que lhes concederia acesso aos dados do utilizador e uma porta de acesso a acções mais gravosas contra o sistema. A escolha da *password* deve por isso ser cuidada, respeitando as seguintes regras:

- Conter pelo menos 8 caracteres;
- Não formar uma palavra ou referência facilmente associável ao utilizador (por exemplo, nome próprio ou login);
- Conter uma mistura de letras (preferencialmente maiúsculas e minúsculas), algarismos e símbolos;

A password deve também ser alterada periodicamente e, como já referimos, não ser concedida a qualquer outro utilizador. Note-se que isto não representa desconfiança para com outra pessoa mas sim o assumir de uma atitude responsável que reflecte a posição a tomar ao longo de toda a vida como profissionais de uma área tão sensível a fugas de informação.

Pelas mesmas razões, todos os utilizadores deverão assegurar-se que não abandonam as máquinas (por qualquer período de tempo) sem proceder ao encerramento da sessão (logout). O sistema poderá interromper esta operação por diversas razões, por exemplo, se o utilizador não salvaguardou um ficheiro. Chama-se por isso a atenção para a necessidade de se assegurar que a sessão foi realmente terminada, o que normalmente acontece com a reapresentação do écran de login ou, eventualmente, com a reinicialização da máquina.

Capítulo 2

O Linux Loader

Os computadores pessoais dos laboratórios têm disponíveis dois sistemas operativos: o Windows NT e o Linux, dos quais falaremos mais adiante. No entanto, não é possível executá-los em simultâneo.

Assim, quando um computador é iniciado¹ o utilizador pode escolher qual dos sistemas operativos pretende executar através do Linux Loader, vulgarmente denominado LILO.

Quando o LILO é executado, surge no écran a seguinte mensagem:

```
LILO
```

Se durante algum tempo o utilizador não premir uma das teclas Shift, Alt ou Ctrl o LILO selecciona o sistema operativo por omissão (à data, o Windows NT). Caso contrário surge a mensagem:

```
LILO boot:
```

Neste ponto, a tecla <TAB> apresenta os sistemas operativos disponíveis. Para seleccionar um, digita-se o seu nome e prime-se Enter.

Exemplo:

```
LILO
<ALT>
LILO boot:  <TAB>
NT Linux
LILO boot:  Linux <ENTER>
```

Nota: a utilização de <TAB> é desnecessária se o nome do sistema operativo for conhecido.

Começará então o processo de inicialização do sistema operativo seleccionado.

¹Do que falaremos mais adiante

Capítulo 3

O Windows NT

O Microsoft Windows NT for Workstation v. 4.0 é um dos sistemas operativos disponíveis nos laboratórios. Dispõe de uma interface com o utilizador semelhante à dos restantes sistemas operativos da Microsoft, nomeadamente: o Windows 95 e Windows 98.

3.1 Login e Logout no Windows NT

A janela de validação de utilizadores no Windows NT tem três campos de diálogo:

User name Identificação do utilizador atribuída aquando da abertura de áreas pela administração.¹

Password sequência de caracteres conhecida unicamente pelo utilizador e que comprova a sua identidade. Os caracteres digitados são apresentados no écran com asteriscos (*).

Domain Conjunto lógico de máquinas onde o utilizador está registado e em função do qual pretende ter acesso ao sistema. Presentemente, o domínio onde os alunos estão registados é INFORMATICA.

Após o preenchimento correcto destes três campos, o botão Ok ou a tecla Enter procederão à validação do utilizador no sistema. Se estiverem correctos será concedida entrada no sistema. Se algum deles estiver incorrecto, o sistema informará o utilizador através de uma janela e voltará a apresentar o menu inicial. O problema mais usual é a digitação de uma password incorrecta. Uma vez que a password é *case-sensitive*, isto é, distingue maiúsculas de minúsculas, verifique se a tecla Caps Lock não está activada.

A saída do Windows NT é feita pela opção *Shut down* do menu *Start*. Surge então uma janela onde o utilizador selecciona o tipo de operação que pretende executar (Figura 3.1), nomeadamente:

¹Presentemente a política de atribuição de nomes de utilizador é ixxxxx para os alunos da licenciatura em Informática e exxxxx para os alunos da licenciatura em Engenharia da Linguagem e do Conhecimento onde xxxxx é substituído pelo número de aluno.

- Desligar o computador: que pressupõe que o computador seria desligado em seguida;
- Reinicializar o computador: que faz reset ao sistema, surgindo novamente o LILO;
- Encerrar a sessão: que faz surgir novamente a janela de entrada na sessão.

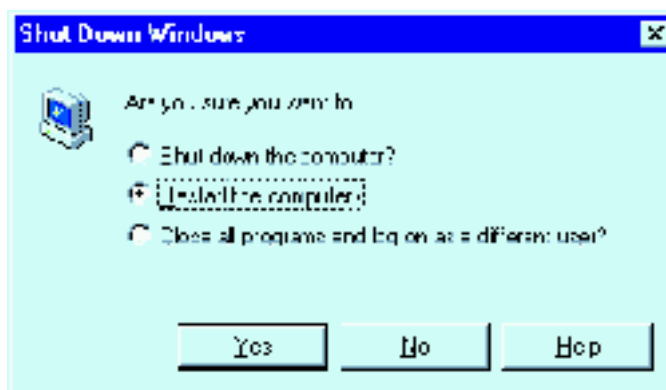


Figura 3.1: Janela de encerramento de sessão do sistema operativo Windows NT

Em qualquer uma delas o utilizador terá a sua sessão encerrada e necessitará de reiniciar o processo se pretender retomar o trabalho.

3.2 Persistência dos Dados

Com a configuração actual, os dados colocados por um utilizador numa máquina em Windows NT são:

- Acedidos novamente apenas nessa máquina;
- Visíveis e alteráveis por todos os utilizadores

Chama-se por isso a atenção que:

- Todos os trabalhos realizados em Windows NT devem ser guardados numa diskette do utilizador;
- Antes de sair do NT todos os utilizadores deverão apagar os ficheiros produzidos, evitando assim:
 - O acesso por terceiros aos seus trabalhos;
 - A acumulação de “lixo” nas máquinas;

3.3 Utilização do software

Algumas aplicações necessárias para as cadeiras não foram instaladas em todas as máquinas e encontram-se concentradas num servidor podendo ser obtidas a partir dele.² Para aceder a essas aplicações, os utilizadores deverão:

1. Abrir o ícone Network Neighborhood;
2. Seleccionar o servidor pretendido;
3. Navegar pelas directorias apresentadas até encontrar o executável pretendido;
4. Executá-lo.

Note-se que, embora executado localmente, este processo não implicou uma transferência de ficheiros para a máquina local.

3.4 Mudar do sistema operativo Windows NT para Linux

A mudança de sistema operativo é sempre feita através da reinicialização da máquina por forma a aceder novamente ao LILO. Esta reinicialização **não** implica o desligar da máquina. Se não estiver uma sessão aberta, o utilizador deverá abrir uma para executar o processo de *shutdown* descrito na secção 3.1.

Caso exista uma sessão aberta, o utilizador deve aceder à janela apresentada na Figura 3.1 (pág. 8) e reinicializar o computador.

Atenção: Não desligar o computador sem que o sistema operativo o indique explicitamente. Desligar o computador com a janela de login visível poderá originar perdas de dados.

²Presentemente, a máquina que presta este serviço chama-se *bit*.

Parte II

O Sistema Operativo Unix

Capítulo 4

Unix e Linux

Nota: Algum do conteúdo da Parte II foi retirado do trabalho “Introdução à utilização do Sistema UNIX” [PDMM90].

O Unix é um sistema operativo criado no início da década de 70 nos Bell Lab’s nos Estados Unidos. É um sistema multi-utilizador desenvolvido na linguagem de programação C. Actualmente existem diversas versões (por exemplo, BSD Unix, System V, HPUX, Solaris), todas elas orientadas para médios e grandes sistemas.

O Linux é uma versão do Unix para pequenos computadores desenvolvida por Linus Torvalds. Mantendo todas as características do Unix, é orientado para computadores de pequena dimensão, estando disponíveis versões para máquinas da família Intel, Alpha, etc. Para além das possibilidades que oferece, o Linux tem-se expandido mundialmente por ser de distribuição gratuita. Actualmente, o Linux é mantido por um conjunto de programadores que de forma gratuita o vão actualizando. Presentemente existem diversas versões disponíveis, por exemplo, Red Hat, Slackware e Debian. A opção por uma prende-se normalmente com razões culturais embora algumas apresentem vantagens sobre as restantes. Por exemplo, o Red Hat é conhecido pela facilidade de configuração e adição de software.

Um dos grandes distribuidores de software para Linux é a FSF (Free Software Foundation) com o seu projecto GNU. São deles as mais populares versões de compiladores (gcc), debuggers (gdb), editores de texto (gnu emacs) e utilitários de compressão (gzip).

Na prática, qualquer pessoa pode contribuir para o desenvolvimento do Linux produzindo e distribuindo software. Grandes outras marcas têm disponibilizado software de forma gratuita para este sistema operativo.

Capítulo 5

Conceitos elementares

Ao contrário do Unix, orientado para sistemas centralizados, onde uma única máquina serve diversas estações de trabalho, o modo mais natural de operação do Linux é a execução local do sistema operativo em cada estação de trabalho (computador pessoal). Quando o sistema operativo é seleccionado (ver LILO na pág. 5), imprime um conjunto de mensagens no écran que indicam os módulos que se encontra a carregar e o resultado de alguns testes básicos ao sistema. Quando o carregamento do sistema operativo estiver concluído, as máquinas darão, em princípio, início ao ambiente gráfico X Windows (vulgarmente apenas designado por X), do qual, por não fazer parte do sistema operativo falaremos mais à frente (ver Parte III, pág. 43).

O Linux disponibiliza sessões independentes em simultâneo. Por sessão entende-se um ambiente de trabalho completamente independente dos restantes onde o utilizador tem que se registar e sair. As sessões podem ser comutadas entre si pela utilização de Alt+Fn onde n é o número da sessão pretendida. O X Windows ocupa normalmente a sessão 7. Para obter uma sessão sem este ambiente gráfico (designada *sessão em modo de texto*), basta seleccionar qualquer uma das restantes. Por exemplo, a sessão 1 é obtida por Ctrl+Alt+F1.¹

5.1 Persistência dos dados

Ao contrário do que acontece com os dados colocados pelo utilizador no Windows NT, no Linux os dados colocados na sua área individual estarão disponíveis em qualquer outra máquina em que o utilizador faça login. Isto porque todas as máquinas têm instalado um sistema de ficheiros distribuído² denominado NFS (Network File System). Assim, apesar de aparentemente estarem guardados na máquina onde o utilizador se encontra, os ficheiros são colocados numa outra máquina que os distribui por todos os PC's quando estes os solicitam.³ Na eventualidade de esta máquina não se encontrar disponível, todos os ficheiros ficarão inacessíveis, independentemente da máquina onde foram colocados. Um exemplo de utilização é apresentado na figura 5.1.

¹Note-se que entre sessões em modo de texto a combinação é Alt+Fn mas para sair da sessão em X Windows é também necessária a utilização de Ctrl.

²Sistemas de ficheiros distribuídos serão matéria de uma das cadeiras do curso.

³Presentemente esta máquina chama-se caravela.

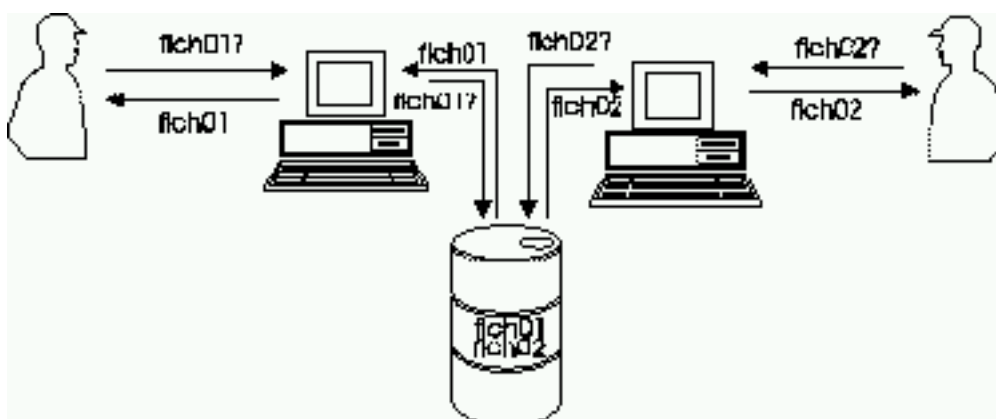


Figura 5.1: Funcionamento de um sistema de ficheiros distribuído

Um sinal da indisponibilidade do servidor de ficheiros é a mensagem de erro: `Could not chdir to home directory /home/xpto: No such file or directory.` quando o utilizador faz o login na máquina. Mesmo que o servidor se torne entretanto disponível, será necessário reinicializar o sistema para voltar a ter acesso aos ficheiros pessoais.

5.2 Entrada e saída de uma sessão em Linux

A entrada numa sessão em Linux é feita pela digitação de um par (login,password). O login é atribuído pela administração do sistema e é permanente. A password é inicialmente atribuída pela administração mas deverá ser alterada logo que possível e com regularidade, por forma a respeitar os requisitos indicados na secção 1.1. O sistema começa por solicitar o login do utilizador. A digitação deverá ser concluída com Enter. Erros de digitação poderão ser corrigidos com backspace. Em seguida o sistema solicitará a password.

Esta não será visível no écran. Embora seja possível, a sua correcção com backspace torna-se difícil uma vez que não são visíveis os caracteres a apagar. Se o login ou a password estiverem incorrectos, surgirá a mensagem:

Login incorrect

As causas mais comuns de erros de login são:

- Nome de login inválido, eventualmente poderá ter sido digitado o login de outro sistema;
- Password inválida por erro de dactilografia
- Caixa de login e/ou password incorrecta. O sistema distingue maiúsculas de minúsculas. Deve-se verificar se a tecla de Caps Lock não se encontra activada.

Após uma validação com sucesso o sistema executará os ficheiros apresentados mais à frente neste documento.

A saída de uma sessão será feita pelo comando `logout`, `exit` ou ainda, nalgumas configurações por `^D`. Após a terminação da sessão surgirá novamente o diálogo de abertura de sessão.

Atenção: É frequente ao longo de um dia de trabalho os utilizadores abrirem várias sessões na mesma máquina. Se for esse o caso, assegure-se do encerramento de todas elas percorrendo-as (`Alt+F1` a `Alt+F7`) antes de abandonar a máquina.

5.3 Reinicialização de um sistema Linux

O reinício da máquina (para obter novamente o LILO) é feito premindo simultaneamente a combinação `Ctrl+Alt+delete` na consola⁴ quando o sistema se encontra numa das sessões em modo de texto. Este procedimento deverá ser executado quando todas as sessões estiverem encerradas sob pena de ocorrerem perda ou danificação dos ficheiros com que o utilizador trabalhou.

Surgirão então um conjunto de mensagens indicando os procedimentos que o sistema está a tomar para essa operação.

5.4 Sequências de controlo

O Unix, à semelhança de muitos outros sistemas utiliza um conjunto de chaves ou controlos especiais que permitem ao utilizador assinalar certas decisões e situações importantes durante o diálogo com o utilizador. As principais sequências de controlo estão listadas na tabela 5.1

<code>^U</code>	Cancelamento de uma linha de comando. A totalidade da linha é eliminada, sendo de novo apresentado o prompt do interpretador de comandos.
<code>^C</code>	Terminar a execução de um programa ou de qualquer comando em execução. Pode não sortir efeito uma vez que alguns programas estão preparados para ignorar esta sequência. Caso a execução não termine tente ainda <code>q</code> seguido de <code><Enter></code> , <code>^D</code> e <code>^Y</code> .
<code>^S/^Q</code>	Suspender o scroll do écran e voltar a activá-lo. Em alternativa deve ser usado um dos filtros <code>less</code> ou <code>more</code> .

Tabela 5.1: Sequências de controlo do Unix

5.5 Letras maiúsculas e minúsculas

O interpretador de comandos do Unix é sensível ao tipo de caixa da letra nos identificadores de ficheiros, comandos e opções.

Se por um lado esta distinção representa uma maior liberdade na atribuição de nomes a ficheiros e comandos, por outro ela está muitas vezes na base de

⁴isto é, no teclado da máquina onde o sistema operativo está a ser executado e não num qualquer terminal que lhe esteja eventualmente ligado.

alguns erros cometidos, em especial, por utilizadores com experiência prévia de outros sistemas operativos.

Assim, por exemplo, os identificadores de ficheiros “xpto.c” e “Xpto.c”, designam dois ficheiros distintos, e a opção “l” em `ls -l` não produz o mesmo efeito que a opção “L”.

A maioria dos comandos em Unix está definida em caixa baixa (minúsculas).

5.6 Administrador de sistema

Todos os sistemas Unix têm um utilizador com poderes especiais denominado *root*.

O root é o único utilizador que tem permissão para:

- Adicionar e eliminar utilizadores do sistema;
- Mudar a password de qualquer utilizador;
- Utilizar qualquer ficheiro independentemente das restrições de acesso estabelecidas (abordadas mais à frente neste documento);
- Matar qualquer processo;
- Terminar e reinicializar o sistema operativo;⁵
- Configurar o sistema;

Se assim o entender, o root pode delegar a execução de algumas destas opções noutros utilizadores os quais no entanto, actuarão em seu nome.

5.7 Ficheiros

O conceito de ficheiro é bastante importante em Unix. Quase todos os repositórios de informação, sejam eles textos legíveis, sequências de caracteres correspondentes ao resultado de um programa, etc., são ficheiros.

Existem fundamentalmente dois tipos de ficheiros, a saber, *ficheiros simples* e *directorias*. Começaremos por descrever o primeiro tipo de ficheiros.

5.7.1 Nomes dos ficheiros

Quando é criado um ficheiro é-lhe atribuído um nome, o qual passará a constituir a identificação desse ficheiro a nível da directoria onde é criado.

Em Unix, o nome de um ficheiro pode ter até 256 caracteres de comprimento.⁶ Podem ser utilizados quaisquer caracteres excepto a barra (`/`), embora seja conveniente restringir essa utilização aos caracteres alfanuméricos (letras e algarismos) e ainda aos caracteres “_” (underscore) e “.” (ponto).

⁵No Linux esta opção é também dada aos outros utilizadores através da sequência `Ctrl+Alt+delete` já referida

⁶Em algumas versões mais antigas do Unix, este valor poderá estar reduzido a 13

Convenções de designação dos ficheiros É vulgar encontrar em Unix o nome dos ficheiros dividido em duas partes pelo carácter “.”. A primeira parte identifica o ficheiro como elemento de uma dada classe de ficheiros, e a segunda, é a *extensão*, ou a classe do ficheiro.

Alguns comandos do Unix são restritivos quanto às extensões que esperam nos ficheiros com que lidam. Os compiladores são disso um bom exemplo. Assim, qualquer programa escrito na linguagem C deverá ter a extensão “c” para ser compilado, enquanto que, por exemplo para o Pascal a extensão é “p”.

A definição de extensões facilita de certa forma a consulta de ficheiros de um catálogo. Por exemplo, é usual utilizar a extensão “h” em ficheiros nos quais estejam definidas estruturas a incluir na compilação de um dado programa. Outras extensões usuais são “txt” para ficheiros de texto, “bas” para programas em basic, etc.

Notas:

1. Ao contrário do MS-DOS, o sistema operativo Unix não faz interpretação das extensões dos ficheiros. Um ficheiro é, aos olhos do sistema operativo, executável se o utilizador detiver permissões para tal e não se contiver uma dada extensão. Tipicamente, no UNIX um ficheiro executável não possui extensão.
2. Em algumas situações é usual um ficheiro conter mais que uma extensão. A utilização do carácter “.” não sofre de qualquer restrição, podendo ocorrer diversas vezes num mesmo ficheiro. O número de caracteres após este também não é limitado.

Alguns ficheiros especiais começam com o “.” (só têm extensão). É o caso de um ficheiro associado a cada área de trabalho em Unix (`.profile`), cujo conteúdo é executado sempre que é efectuada uma entrada nessa área.⁷ Este tipo de ficheiros tem a particularidade de não ser visível na listagem de ficheiros de uma directoria (comando `ls`) excepto quando é utilizada a opção `-a`.

5.7.2 Meta-caracteres

Na identificação de ficheiros acontece muitas vezes estarmos interessados em:

- Encontrar todos os ficheiros com uma mesma extensão
- Encontrar todos os ficheiros em cujos nomes figure uma dada sequência de caracteres
- Encontrar todos os ficheiros cujo nome, ou a extensão, tem um dado comprimento de caracteres, etc.

Para este tipo de pesquisa é possível utilizar dois “meta-caracteres”⁸, tais que:

* Substitui qualquer sequência de caracteres;

⁷O nome deste ficheiro varia com a shell utilizada, por exemplo, na bash o ficheiro chama-se `.bash_profile` e na cshell `.login`

⁸em inglês *wildcards*

? substitui um caracter qualquer;

Os *wildcards* podem ser aplicados em qualquer comando que suporte a utilização de mais que um nome de ficheiro. Por exemplo, os comandos de listar directoria (`ls`) e eliminar ficheiro(s) (`rm`).

Exemplos

*.f identifica todos os ficheiros numa directoria com extensão f;

aaa*r1.c identifica todos os ficheiros numa directoria iniciados por aaa e terminados por r1.c;

*.??? identifica todos os ficheiros numa directoria cujo comprimento da extensão é exactamente de 3 caracteres;

5.8 Directorias

As directorias constituem um tipo especial de ficheiros. A sua função é agrupar ficheiros e outras directorias. Estes dir-se-ão *contidos* naquela.

Uma *subdirectorias* é um abuso de linguagem utilizado para designar uma directoria que se encontra dentro de uma outra.

No que se segue, as características de uma directoria enquanto ficheiro Unix serão esquecidas, sendo alvo da nossa atenção apenas o conceito (lógico) de directoria que será necessário para o utilizador do sistema.

Todos os ficheiros no Unix são guardados logicamente segundo uma estrutura hierárquica em árvore.

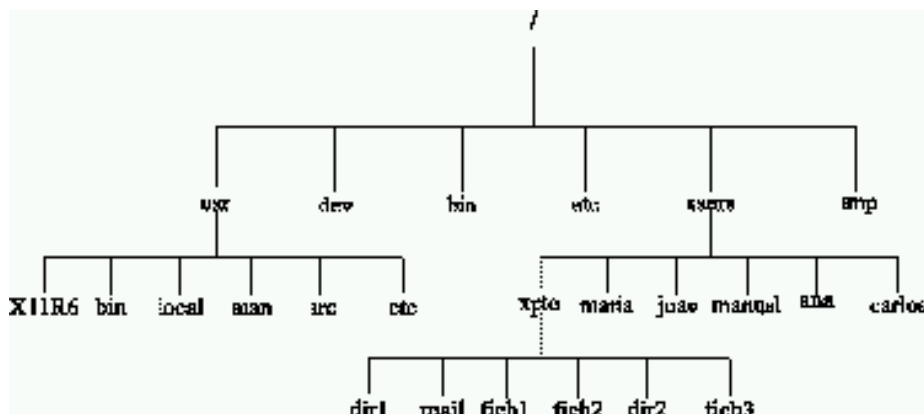


Figura 5.2: Exemplo de uma árvore de directorias

Na figura 5.2 salienta-se:

1. A árvore tem uma única *raiz*, identificada por “/”. Este é o *nome* da raiz de qualquer sistema de ficheiros Unix. Todos os ficheiros e directorias são colocados abaixo da raiz.

2. Na directoria `xpto` existem ficheiros *regulares* e outras directorias. Qualquer directoria pode conter ambos os tipos de ficheiros.
3. Existem duas directorias chamadas `etc`. Dois ficheiros (regulares ou directorias) podem ter o mesmo nome desde que não se encontrem dentro da mesma directoria.

Apesar de poder haver ficheiros com o mesmo nome no sistema, qualquer ficheiro pode ser univocamente identificado especificando o caminho que se percorre na árvore para o atingir a partir da raiz. O nome da raiz (a barra) deve nesse caso ser incluído. O separador entre nomes é também a barra (/). Por exemplo, o ficheiro `fich1` da figura 5.2 tem o seguinte nome:

```
/users/xpto/fich1
```

enquanto que o nome completo da directoria `local` é:

```
/usr/local
```

À identificação completa de um ficheiro dá-se o nome de *pathname* ou simplesmente *path*.⁹

Nota: O comando `ls` que será descrito em pormenor mais adiante lista o conteúdo de uma directoria. Como exercício sugere-se a utilização deste comando com diversos caminhos (por exemplo `ls /`) para testar estes conceitos.

5.8.1 Directoria corrente

Durante uma sessão no sistema, todos os utilizadores têm uma directoria denominada *directoria corrente* vista como a directoria onde o utilizador se encontra nesse momento.¹⁰ A directoria corrente é aquela em que os ficheiros nela contidos são identificados apenas pelo seu nome. Normalmente, o *prompt* indica a directoria corrente.¹¹ Por exemplo, o *prompt*

```
[xpto@di-3 dir1]$
```

indica que o utilizador `xpto` se *encontra* na directoria `dir1`.

Qualquer ficheiro pode também ser identificado pelo caminho a partir da directoria corrente omitindo a / inicial no *pathname*. Para especificar a directoria acima da corrente utiliza-se o nome de *ficheiro* “`..`” (dois pontos finais consecutivos). Para referir uma directoria abaixo utiliza-se o seu nome. A directoria corrente é identificada pelo nome de *ficheiro* “`.`” (ponto final).¹² Qualquer ficheiro pode portanto ser identificado univocamente de duas formas:

1. Especificando o seu *pathname* completo (a partir da raiz).
2. Especificando o seu nome a partir da directoria corrente.

⁹Que em português significa *caminho*

¹⁰Supondo que existe a possibilidade de o utilizador se posicionar numa directoria, o que é uma simplificação do sistema.

¹¹A mudança de directoria corrente é feita pelo comando `cd` que será apresentado mais à frente.

¹²Embora aparentemente desnecessária, esta identificação pode ser útil quando queremos indicar ao sistema um nome qualificado para um ficheiro na directoria corrente. Por exemplo `./fich1`

Exemplos: Retomando novamente o exemplo da figura 5.2 e supondo que a directoria corrente era `/users` temos que os caminhos:

`xpto/fich1`, `/users/xpto/fich1` e `./xpto/fich1` Identificam o ficheiro `fich1` dentro da directoria `xpto`.

`/usr/man` e `../usr/man` Identificam ambos a directoria `man` dentro da directoria `usr`.

`maria`, `./maria` e `/users/maria` Identificam a directoria `maria` existente dentro de `users`.

`/etc` e `/usr/etc` Identificam duas directorias distintas. A primeira poderia também ser referida por `../etc` e a segunda por `../usr/etc`.

As *shells* modernas dispõem de uma facilidade de apoio à composição de *pathnames* no *prompt*. A tecla <TAB> tenta sempre completar o texto já digitado por forma a gerar um *pathname* válido.

Exemplo: Tomando a árvore da figura 5.2 (pág. 20) e tendo o utilizador digitado:

```
[xpto@di-3 dir1]$/u
```

e primindo em seguida <TAB>, a *shell* completa o *pathname* resultando

```
[xpto@di-3 dir1]$/us
```

Uma vez que as duas directorias da raiz começadas por “u” são seguidas por um “s”. Se o utilizador pretender saber quais as hipóteses de preenchimento que geram um *pathname* válido poderá digitar <TAB> duas vezes consecutivas ao que lhe será apresentado:

```
users usr
```

```
[xpto@di-3 dir1]$/us
```

A desambiguidade poderá ser retirada pelo utilizador através da digitação de uma letra. Por exemplo, acrescentando a letra “e” ao *pathname* em construção, ou seja, ficando com:

```
[xpto@di-3 dir1]$/use
```

e primindo novamente <TAB> o sistema responde com:

```
[xpto@di-3 dir1]$/users/
```

Este processo pode continuar ao longo de toda a árvore e não necessita de ser iniciado na raiz. Tal como em todas as restantes operações do sistema, aplicam-se as restrições impostas pelas permissões de leitura e execução para os ficheiros e directorias especificados.

5.8.2 A directoria HOME

O Unix guarda numa variável denominada `HOME` a identificação da directoria de “residência” de cada utilizador, o qual corresponde ao início da sub-árvore onde ele guarda os seus ficheiros individuais.¹³

Um utilizador pode referir a sua própria directoria `HOME` através do caracter especial “~” (til) ou da expressão `$HOME` e o catálogo de um outro utilizador do sistema pela sequência “~ utilizador”. Esta referência pode ser utilizada em qualquer nome completo de ficheiro e em qualquer comando.

¹³Este é o procedimento usual, a política de administração de alguns sistemas poderá levar a que sejam utilizados outros métodos.

Capítulo 6

Execução de comandos

6.1 Os interpretadores de comandos do Unix

Após a entrada numa área de trabalho, a sessão de trabalho está iniciada. Daqui em diante o sistema passará a comunicar com o utilizador através de um programa especial (denominado interpretador de comandos ou *shell*) cuja função é receber as directivas do utilizador (comandos) e lançar a sua execução. Terminada a execução de cada comando, o programa interpretador de comandos volta a indicar a sua disponibilidade apresentando o *prompt* no início de cada nova linha.

Nas próximas secções será assumido que todos os exemplos de diálogos com o utilizador serão efectuados na área *xpto* da máquina “di-3”, cujo prompt é `xpto@di-3>`.

O Unix dispõe de diferentes interpretadores de comandos (shells). Por exemplo `bash` (Bourne Again Shell), `shell`, `csh` (C-shell) e `tsh` (TC-shell). As diferenças para os utilizadores principiantes são mínimas e praticamente irrelevantes pelo que não serão descritas neste documento. Sugere-se a consulta das páginas de manual adequadas para uma descrição mais profunda.

6.1.1 Variáveis de shells

Tal como as linguagens de programação, as shells também dispõem de ferramentas que permitem associar identificadores a valores, os quais, são também denominados por variáveis. Um caso particular destas são as variáveis de ambiente, as quais controlam alguns parâmetros do ambiente de execução. São exemplos destas, as variáveis `HOME` e `PATH` que indicam respectivamente a directoria corrente do utilizador aquando do *login* e o conjunto de directorias pesquisáveis em busca dos ficheiros de comandos.

Listas de entradas numa variável de ambiente são separadas por “:”. Por exemplo, um valor possível para a variável `PATH` é:

```
/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:.
```

Que lista as 5 directorias onde serão pesquisados os comandos digitados pelo utilizador.

As operações que se podem executar sobre as variáveis são semelhantes às das linguagens de programação: definição, alteração e eliminação. As variáveis de shell são definidas com o tipo “cadeias de caracteres” (*strings*).

A forma de realizar as operações sobre as variáveis varia ligeiramente com a shell adoptada. Por ser a mais utilizada, nesta secção apresentam-se os comandos da **bash** que realizam as operações sobre variáveis.

São definidos dois conceitos. VAR representa o “nome da variável” enquanto que “\$VAR” representa o “conteúdo da variável”. Por convenção os nomes das variáveis são sempre em maiúsculas.

A definição de uma variável é feita digitando o comando

```
NOMEVAR=valor
```

Alterações ao seu valor são feitas da mesma forma que as definições, uma vez que não pode haver duas variáveis com o mesmo nome. Um caso particular de alteração é a concatenação ao valor já existente. Para tal, utiliza-se o conceito de “conteúdo da variável”. Por exemplo, para adicionar a directoria `/users/xpto/mybin` ao fim da lista da variável `PATH` executar-se-ia:

```
PATH=$PATH:/users/xpto/mybin
```

Note-se que a adição ao início da lista seria conseguida invertendo a ordem de especificação dos valores.

A remoção de uma variável de ambiente é feita pelo comando `unset VAR`. A lista de variáveis definidas pode ser visualizada pelo comando `set`.

Quando uma variável de ambiente é definida, o seu âmbito restringe-se à shell em que está a executar e não a shells ou processos filhos que venham a ser criados no seu âmbito. O comando `export VAR` propaga a definição da variável a todos os processos filhos que venham a ser criados.

6.2 Formato dos comandos

O formato genérico dos comandos em Unix é:

```
comando [opcoes] {ficheiros}
```

As três partes constituintes de um comando devem ser separadas por espaços em branco ou por caracteres `<TAB>`.

Comando Os comandos Unix são todos identificados por sequências de caracteres iniciados por uma letra, e onde todas as letras são minúsculas. Os comandos Unix não são mais que nomes de programas cuja invocação é desencadeada sempre que são fornecidos ao interpretador de comandos.

O interpretador de comandos não reconhece abreviaturas.

Quando o identificador digitado não corresponde a algum comando reconhecido, a shell responde com `command not found`.

A directoria em que o ficheiro de um comando se encontra não necessita de ser especificada se a sua localização constar da lista na variável `PATH` da shell. Note-se que, por omissão, e ao contrário dos sistemas operativos da *Microsoft*, o Unix não procura o ficheiro na directoria corrente. A execução de um comando localizado na directoria corrente, quando esta não consta de `PATH` é conseguida de três formas:

1. Qualificando o nome com a directoria corrente através do carácter “.” (por exemplo `./comando`);
2. Adicionando a directoria corrente a `PATH`.

3. Adicionando o ficheiro especial “.” a PATH. Nesse caso, qualquer que seja a directoria corrente, o sistema procurará sempre os comandos também nessa directoria.

O Unix respeita a ordem das directorias indicadas em PATH na pesquisa dos comandos. De dois comandos com o mesmo nome será invocado aquele cuja localização se encontre primeiro em PATH.

Nota: Este facto pode resultar numa quebra de segurança se de alguma forma for introduzido no PATH do utilizador uma directoria contendo programas alterados. Sem se aperceber, o utilizador poderá estar a executar acções de terceiros em seu nome, concedendo-lhes permissões que eles não detinham.

Opções As opções podem ou não estar presentes e têm por função modificar a forma de execução dos comandos, um pouco como os adjectivos e os advérbios actuam sobre as nossas frases.

As opções são:

Letras Por exemplo, `ls -a`.

Letras seguidas de valores Por exemplo, `lpr -P lp1`

Cada opção é precedida do carácter “-”, e é possível juntar várias opções numa cadeia. Por exemplo, para executar o comando `ls` (listagem de ficheiros) com as opções “l” e “a” poderá ser digitado `ls -la` ou `ls -l -a`.

Devido ao carácter ambíguo com que os valores associados a opções são interpretados, é conveniente que este tipo de opção seja sempre especificado em separado, por exemplo: `-x -G valor` em vez de `-xG valor`.

Se é cometido um erro na especificação das opções de um comando, o respectivo programa indica qual a sintaxe correcta.

Ficheiros Os ficheiros especificados num comando são aqueles sobre os quais se pretende que esse programa actue. Nem todos os comandos Unix aceitam a especificação de ficheiros.

Os nomes de ficheiros podem, em qualquer situação, ser indicados de forma simples (relativa à directoria corrente) ou qualificada (com indicação da directoria em que se encontra). Este conceito está descrito na secção 5.8 (pág. 20).

Capítulo 7

Comandos do Unix

7.1 Modificação da password

Como já foi referido, cada área tem associada uma password cuja inicialização por defeito, aquando da criação da área, pode e deve ser alterada pelo utilizador. É importante salientar que após a redefinição da password, apenas o responsável por essa alteração ficará conhecedor de qual ela é. Nem o próprio administrador do sistema a conhecerá.

As passwords em Unix são sequências de caracteres podendo ser compostas por quaisquer símbolos alfa-numéricos.

O comando que permite a redefinição de uma password é o comando `passwd`. Para que um utilizador possa alterar a password terá que digitar a anterior.

Notas:

1. Apesar de não ter conhecimento das passwords, o administrador de sistema pode alterar a password de qualquer utilizador.
2. Alguns sistemas operativos dispõem de uma ferramenta de validação que verifica a *qualidade* das passwords propostas desaconselhando as mais evidentes.

7.1.1 Exemplo de utilização

O comando `passwd` faz a definição da nova password em 3 fases:

1. Validação da password existente. O sistema assegura-se da identidade do utilizador que se encontra a realizar a operação.
2. Digitação da nova password.
3. Re-digitação da nova password. Para garantir que não houve erros de digitação uma vez que as passwords não são ecoadas no écran.

Um exemplo de mudança de password é apresentado em seguida:

```
xpto@di-3> passwd <Enter>
Changing password for xpto
```

```
(current) UNIX password: (Digitação da password actual) <Enter>
New UNIX password: (Nova password) <Enter>
Retype new UNIX password: (re-digitação da nova password) <Enter>
```

Se não houver coincidência entre as duas digitações da nova password, o sistema apresentará a mensagem de erro `Sorry, passwords do not match` e reiniciará o processo.

7.2 Manual interactivo

Nos sistemas Unix está disponível um manual interactivo ou *on-line* e que permite obter várias informações referentes à utilização de comandos.

7.2.1 Comando man

O comando `man` permite obter ajuda sobre a invocação de qualquer comando listado no manual de Unix. `man` tem a seguinte sintaxe:

```
man [opcoes] comando
```

Se houver informação respeitante a `comando`, `man` apresentará no écran o texto dessa informação dividido em três secções (tal como no manual de Unix).

A secção *Name* descreve sumariamente o comando. Na secção *Synopsis* é apresentada a sintaxe de invocação. Finalmente a secção *Description* explica com maior detalhe o comando, descrevendo os seus argumentos, opções e ficheiros. Uma secção opcional de grande interesse é *See also* que encaminha o utilizador para as páginas de manual de funções relacionadas.

Para facilitar a leitura, toda a informação é apresentada página a página. O avanço nas páginas é feito pela barra de espaços. A terminação de visualização e opções de recuo na página está dependente do filtro de controlo de visualização utilizado (`more`, `less` ou nenhum).

As páginas de manual estão organizadas em secções. Alguns comandos (por exemplo, `printf`) dispõem de mais de uma página de manual, dispersas por várias secções. Por omissão, `man` executa a pesquisa apenas até localizar a primeira ocorrência para o comando. Este comportamento pode ser alterado recorrendo a duas opções:

- A opção `-a` pesquisa em todas as secções a página referente ao comando. Se existir mais que uma página, a seguinte será apresentada quando a visualização da anterior for terminada (não por `^C`).
- A opção `-S` aceita como argumento um conjunto de secções separadas pelo caracter dois pontos (:). Pode-se por isso fazer `man -S 3 printf` para visualizar a página referente ao comando `printf` da secção 3 em vez da referente ao comando na secção 1 que seria apresentada por omissão.

Presentemente, grande parte dos sistemas estenderam esta ferramenta às linguagens de programação para que estão instalados compiladores. No Linux, é muito comum que as funções da linguagem C disponham também de páginas de manual.

7.2.2 Comando `apropos`

O comando `man` pressupõe que o utilizador conhece o nome do comando sobre o qual quer obter informação. O comando `apropos` tem um âmbito de aplicação mais largo ao procurar a palavra fornecida como argumento numa base de dados de comandos e suas descrições. Assim, `apropos` apresenta uma lista de comandos e suas descrições (semelhante à secção *Name* do comando `man`) em que a palavra indicada figura ou no nome do comando ou na sua descrição.

Exemplo:

```
[xpto@di-3 xpto]$ apropos search
apropos (1)          - search the whatis database for strings
badblocks (8)       - search a device for bad blocks
bsearch (3)         - binary search of a sorted array.
conflict (8)        - search for alias/password\index{password} conflicts
find (1)            - search for files in a directory hierarchy
hcreate, hdestroy, hsearch (3) - hash table management
lfind, lsearch (3)  - linear search of an array.
lkbib (1)           - search bibliographic databases
lookbib (1)         - search bibliographic databases
lsearch (n)         - See if a list contains a particular element
manpath (1)         - determine user's search path for man\index{man@\texttt{man}} pages
strpbrk (3)         - search a string for any of a set of characters
strspn, strcspn (3) - search a string for a set of characters
tsearch, tfind, tdelete, twalk (3) - manage a binary tree
whatis (1)          - search the whatis database for complete words.
zipgrep (1)         - search possibly zip files for a regular expression
Search::Dict, look (3) - search for key in dictionary file
```

Note-se que `apropos` apresenta todas as ocorrências da expressão `search`, ocorram elas no nome do comando ou na sua descrição. Fica ao critério do utilizador seleccionar qual o comando que pretende e, se necessário, invocar a página de manual adequada.

7.2.3 Comando `whatis`

O comando `whatis` apresenta a mesma informação que `apropos` mas a pesquisa é apenas efectuada ao nome do comando e a palavra terá que corresponder exactamente à dada como argumento.

7.3 Listagem de ficheiros em directorias

Em Unix, o comando para listagem de ficheiros de uma directoria é o `ls`.

O comando `ls` pode ser utilizado na sua forma mais simples (apenas `ls`), ou junto com opções e identificadores de ficheiros.

A utilização do comando `ls` sem parâmetros lista apenas os nomes dos ficheiros, alinhando-os em colunas. Não apresenta qualquer informação adicional (por exemplo se o ficheiro listado é uma directoria) nem os ficheiros iniciados por “.”.

7.3.1 Opções mais comuns

Opção -l A opção -l apresenta um conjunto de informação adicional para cada ficheiro listado. Por exemplo:

```
[xpto@di-3 tmp]$ ls -l
total 4
drwxr-xr-x 2 xpto users 1024 Mar 9 16:38 d1
-rw-r--r-- 1 xpto users 15 Mar 3 13:48 h.txt
-rwxr-xr-x 1 xpto users 5 Mar 9 17:09 myls
-rw-r--r-- 1 xpto users 11 Mar 9 16:39 t.c
```

A informação apresentada pelo comando `ls -l` para cada ficheiro divide-se em 7 campos:

Tipo do ficheiro A primeira posição da sequência de caracteres inicial permite distinguir ficheiros normais de directorias. Se o primeiro caracter for (como no caso de `d1` neste exemplo) 'd', o ficheiro em causa é uma directoria. Um '-' nesta posição indica tratar-se de um ficheiro regular. Um 'l' apresenta um *link simbólico*, conceito que não será abordado neste documento.

Permissões Nos 9 caracteres seguintes a existência de um '-' significa a ausência e um outro caracter a existência de permissão. As permissões em causa são:

r (read) Permissão de leitura do ficheiro

w (write) Permissão de escrita/eliminação do ficheiro

x (eXecute) Permissão de execução do ficheiro

Os primeiros três caracteres descrevem as permissões para o dono¹ do ficheiro, o segundo bloco de três caracteres as permissões para o grupo e as três últimas para os restantes utilizadores.

Estes conceitos são aprofundados na secção 7.7 (pág. 34).

Número de *hard links* Este conceito ultrapassa o âmbito deste documento pelo que não será explicado.

Dono do ficheiro Utilizador que detém a propriedade do ficheiro (*owner*). É este utilizador que desfruta das permissões apresentadas no primeiro bloco. No exemplo, todos os ficheiros são propriedade do utilizador *xpto*.

Grupo do ficheiro Grupo de utilizadores (ao qual pertence o dono do ficheiro) a que o ficheiro está atribuído. São os utilizadores que pertencem a este grupo que desfrutam das permissões indicadas no segundo bloco. No exemplo, todos os ficheiros estão atribuídos ao grupo *users*.

Data/hora da última alteração Em ficheiros alterados há mais de um ano, a hora é substituída pelo ano da última alteração.

Nome do ficheiro Tal como apresentado na execução do comando `ls`.

¹em inglês *owner*

Opção -a A opção `-a` apresenta *todos* os ficheiros da directoria, inclusive os ficheiros iniciados por “.”. Dois ficheiros particulares nesta categoria são os ficheiros “.” e “..” que representam respectivamente a própria directoria e a directoria acima na árvore.

Opção -F A opção `-F` formata o resultado do mesmo modo que o comando `ls` sem opções mas inclui informação sobre o tipo de ficheiro. Assim, as directorias são terminadas por “/” e os ficheiros com permissão de execução por “*”.
Exemplo:

```
[xpto@di-3 tmp]$ ls -F
d1/ h.txt myls* t.c
```

7.4 Operações sobre directorias

7.4.1 Mudança de directoria corrente

Foi já referida a possibilidade de navegação através da árvore de directorias de um sistema Unix. O comando que permite a um utilizador *deslocar-se* para uma nova directoria, a qual passará a ser a sua directoria corrente, é `cd` (*change directory*).

O argumento para `cd` é um qualquer nome de directoria, respeitando as regras enunciadas na secção 5.8 (pág. 20). Para todos os utilizadores, o acesso à sua `HOME` pode ser feito digitando `cd` sem argumentos.²

7.4.2 Determinação da directoria corrente

O comando `pwd` (*Print Working Directory*) envia para o écran o *pathname* completo da directoria corrente.

7.4.3 Criação/eliminação de directorias

Os comandos `mkdir` e `rmdir` criam e eliminam respectivamente a directoria que recebem como argumento e que pode ser especificada sob qualquer uma das formas apresentada na secção 5.8.

Uma nova directoria não poderá ter o mesmo nome que qualquer outro ficheiro existente na directoria onde ela será criada.

Uma directoria só poderá ser apagada se não contiver quaisquer ficheiros ou outras subdirectorias dentro de si.

7.5 Visualização do conteúdo de ficheiros

Para além da utilização de um editor de texto, de entre os quais em Unix existem o *vi*³ e o *emacs*⁴, é possível ler o conteúdo de ficheiros através da invocação dos comandos `cat`, `more` e `less`.

²Para além claro de `cd ~` e de `cd $HOME`.

³Secção 12.1 na pág. 55

⁴Secção 12.3 na pág. 57

7.5.1 Comando `cat`

O comando `cat` *despeja* para o écran a sequência de caracteres que compõe o(s) ficheiro(s) dado(s) como argumento. Se esta sequência ocupar mais que uma página do écran será necessário e regra geral, bastante incómodo, utilizar as sequências `^S` e `^Q` para parar/continuar o preenchimento do écran.

Se não for dado nenhum ficheiro como parâmetro, o `cat` afixa no écran o que for digitado no teclado até à digitação de `^C` (interrupção do comando) ou `^D` (indicação de fim de ficheiro).

7.5.2 Comandos `more` e `less`

Em alternativa ao comando `cat` existem os comandos `more` e `less`. Estes comandos apresentam o conteúdo dos ficheiros dados como argumento de tal forma que o écran é preenchido página a página. A página seguinte é visualizada premindo a barra de espaços. O avanço linha a linha é feito pela tecla `<Enter>`.

Ambos os comandos dispõem de uma facilidade de pesquisa. O padrão a pesquisar é definido premindo a tecla `/` seguida do padrão, concluindo com a tecla `<Enter>`. Ocorrências seguintes do padrão são obtidas utilizando `"n"` em vez da barra de espaços. Os comandos permitem a utilização de padrões de pesquisa muito ricos, inclusive com recurso a meta-caracteres e a expressões regulares complexas. Informação sobre a composição destes padrões e sobre outros comandos pode ser obtida premindo `h` durante a utilização do programa.

O recuo nas páginas visíveis é obtido através do comando `"b"`.

O comando `less` estende as opções de `more`.

7.6 Operações sobre ficheiros

7.6.1 Eliminação de ficheiros

É conveniente eliminar das directorias os ficheiros que já não são necessários, evitando assim uma sobrecarga escusada do sistema. Em particular, deve ser tido em atenção que erros na execução de certos programas dão origem a um ficheiro especial de grandes dimensões denominado `"core"`. Este ficheiro poderá ser útil como auxiliar de depuração de erros mas na maioria dos casos não é utilizado. O sistema operativo notifica o utilizador da criação de um destes ficheiros com a mensagem:

```
Segmentation fault (core dumped)
```

Em Unix, o comando que permite eliminar um ficheiro denomina-se `rm` (remove). Este comando tem por argumento o nome de um ou mais ficheiros. Podem ser utilizados meta-caracteres.⁵

Atenção: O comando `rm` não pede confirmação da operação e não existem utilitários que permitam a recuperação de ficheiros eliminados.

Na sequência do que foi referido na secção sobre templates, a invocação `rm *`

⁵ver secção 5.7.2.

tem a consequência (por vezes catastrófica) de eliminar todos os ficheiros não iniciados por '.' de um ficheiro.

Para a eliminação de directorias pode ser utilizada a opção `-f` embora para essa função exista especificamente o comando `rmdir`, descrito anteriormente.

7.6.2 Cópia de ficheiros

O comando de cópia de ficheiros é o `cp`. O seu formato de invocação é:

```
cp fichOrigem1 [fichOrigem2 [...]] fichCopia
```

Notas:

1. Se `fichCopia` já existir o seu conteúdo original será eliminado;
2. Se `fichCopia` for o nome de uma directoria o ficheiro será copiado para dentro da directoria, mantendo o nome original
3. Se forem utilizados meta-caracteres ou se for indicado mais que um ficheiro de origem, `fichCopia` deverá ser uma directoria. Nesse caso, todos os ficheiros manterão o nome original e serão colocados dentro da directoria indicada.

7.6.3 Alteração do nome de ficheiros

Os nomes de ficheiros podem ser alterados pelo comando `mv` (move). O formato de invocação é:

```
mv nomeAntigo nomeNovo
```

Note-se que este comando permite também mudar ficheiros de directoria, bastando para tal especificar um *pathname* diferente em `nomeNovo`.

À semelhança do comando de cópia, a indicação de um nome de directoria em `nomeNovo` move o ficheiro para dentro da directoria indicada.

O ficheiro indicado em `nomeAntigo` também pode ser o nome de uma directoria. Nesse caso, todo o conteúdo da directoria (incluindo subdirectorias) é movido para a nova localização, dada por `nomeNovo`.

7.6.4 Impressão de ficheiros

Para imprimir um ficheiro será necessário solicitar ao sistema essa tarefa, que a colocará numa fila de espera para impressão. O comando para imprimir ficheiros é o `lpr`.

O ficheiro a enviar para a impressora está dependente das características desta. Se a impressora aceitar directamente a impressão de caracteres ASCII, os ficheiros de texto poderão ser directamente entregues a `lpr`. Se a impressora utilizar outros protocolos (por exemplo *postscript*), os ficheiros poderão ter que ser previamente formatados. Um dos comandos mais utilizados para essa operação é o `mpage`.

A sintaxe do comando é simplesmente:

```
lpr ficheiro [ficheiro [...]]
```

Listagem dos pedidos pendentes de impressão Por vezes torna-se útil verificar qual o estado da fila de espera de uma impressora. O comando que lista a fila de espera de ficheiros aguardando serem impressos é o `lpq`. Os pedidos são identificados por números, os quais deverão ser utilizados para os referir em quaisquer comandos de gestão da fila de espera.

Cancelamento de pedidos de impressão Por vezes poderá acontecer que a listagem impressa de um dado ficheiro já não seja necessária. Para evitar a acumulação de pedidos é conveniente eliminar a respectiva solicitação da fila de espera.

O comando `lprm`, tendo por argumento o número de uma tarefa, procede à sua eliminação da fila de espera para uma impressora. Assim, por exemplo, para eliminar o pedido 122:

```
lprm 122
```

Notas:

1. Não é possível a um utilizador (excepto ao administrador de sistema) eliminar tarefas de outros utilizadores.
2. A opção “-” elimina todos os trabalhos de impressão do utilizador pendentes na fila de espera.

7.7 Protecção de ficheiros

A questão genérica de quem pode fazer o quê com este ou aquele ficheiro é denominada *protecção de ficheiros*.

Em Unix, os utilizadores são divididos em três classes para a determinação dos direitos sobre os ficheiros. Essas classes são:

outros (other) Inclui todos os utilizadores, inclusive utilizadores de outras máquinas.

grupo (group) É um subconjunto de utilizadores definido no ficheiro “/etc/group”. Cada utilizador pode ser membro de vários grupos simultaneamente. Para verificar quais os grupos definidos num dado momento no sistema existe o comando `groups`.

dono (owner) O utilizador que criou ou é proprietário de um dado ficheiro.

Em princípio, o dono de um ficheiro é a pessoa que criou esse ficheiro. É também este utilizador o único que está autorizado (para além do administrador do sistema) a alterar os direitos sobre qualquer dos seus ficheiros.

O grupo de um ficheiro pode ser alterado pelo seu dono através do comando `chgrp` (*change group*).

Cada ficheiro (normal ou directoria) tem três modos de acesso atribuídos a cada uma das classes de utilizadores atrás referidas, nomeadamente, leitura (*read*), escrita (*write*) e execução (*execute*). A interpretação destes modos de acesso é algo distinta entre ficheiros normais e directorias.

Segue-se uma breve descrição dos três modos de acesso:

read Qualquer utilizador com autorização de leitura pode ver o conteúdo de um ficheiro, listando-o por exemplo através dos comandos `cat` e `more` ou utilizando um editor de texto.

No caso de se tratar de uma directoria, o utilizador está autorizado a listar o seu conteúdo através de `ls`. Para obter informação mais detalhada acerca dos ficheiros nele contidos, tal como a que é fornecida por certas opções de `ls`, necessitará porem do direito de execução. A possibilidade de ler o conteúdo dos vários ficheiros da directoria já é associada com os direitos do utilizador em relação a cada um desses ficheiros em particular.

write Um utilizador com o direito de escrita num dado ficheiro pode alterar o seu conteúdo, através por exemplo, de um editor de texto.⁶

No caso de se tratar de uma directoria, o utilizador poderá alterar o seu conteúdo, apagando ou criando ficheiros nela. Tal como acontece com o direito de leitura, o direito de escrita relativamente aos ficheiros de uma directoria é a que estiver definida para cada um dos ficheiros.

execute Um utilizador com direito de execução sobre um ficheiro poderá utilizá-lo a nível do interpretador de comandos como um vulgar comando de sistema, desde que as directorias acima deste possuam autorização de leitura para esse utilizador.

No caso de se tratar de uma directoria, o utilizador pode mudar para essa directoria (comando `cd`) e copiar os seus ficheiros para outras directorias, desde que possua em relação a estas a permissão de escrita.

7.7.1 Verificação dos direitos

Para verificar os direitos sobre um ficheiro utiliza-se o comando `ls -l`. Um ficheiro como todos os direitos atribuídos terá associada a cadeia de caracteres `rw-rw-rw` onde os três primeiros são os direitos referentes ao dono do ficheiro, os segundos ao grupo e os terceiros aos restantes utilizadores. Quando um direito não está disponível, o respectivo símbolo (r, w ou x) é substituído por um “-”.

7.7.2 Representação numérica

Os direitos possuem também uma representação numérica na base 8 (octal). O utilizador mais experimentado adapta-se por vezes melhor a este segundo modo de representação dos direitos.

A cada direito corresponde um símbolo octal, sendo as equivalências definidas na tabela 7.1:

Assim, por exemplo, a “`rw-r-xr--`” (todos os direitos para o criador, leitura e execução para o seu grupo e leitura para os restantes) corresponderá o número:

⁶Alguns editores de texto analisam as permissões do ficheiro abrindo-o o por exemplo em modo só de leitura (*read-only*) se o utilizador não detiver permissões de escrita.

user/read	400
user/write	200
user/execute	100
group/read	40
group/write	20
group/execute	10
other/read	4
other/write	2
other/execute	1

Tabela 7.1: Protecções de ficheiros e valor octal correspondente no sistema Unix

	400
	200
	100
	40
	10
+	4
<hr/>	
	754

7.7.3 Modificação dos direitos

O dono de um ficheiro pode alterar as protecções desse ficheiro através do comando `chmod` (*change mode*), cuja forma geral de invocação é:

```
chmod modo {ficheiro}
```

Este comando não aceita protecções expressas sob a forma de cadeias de caracteres “`rwX`”. Pode utilizar-se o seu equivalente numérico (na forma apresentada na secção anterior) ou então o modo pode ser alterado, relativamente a cada classe de utilizadores, em separado, através de sequências do tipo:

```
classe[+/-]direito{,classe[+/-]direito}
```

onde `classe` não necessita de ser especificada no caso de se aplicar a todos os utilizadores e pode ser uma cadeia com os seguintes caracteres:

`u` para *user*

`g` para *group*

`o` para *others*

`a` para todas as classes (*all*)

`direito` é uma cadeia formada pelos caracteres “`r`”, “`w`” e “`x`”. A colocação do operador `+` significa a concessão das permissões indicadas e o operador `-` a sua omissão.

Exemplos:

`chmod 754 fich1` Especifica para `fich1` as permissões definidas no exemplo da secção anterior;

`chmod +x fich1`, `chmod a+x fich1` e `chmod ugo+x fich1` Concedem todos permissão de execução do ficheiro `fich1` a todos os utilizadores;


```
chmod ug+rx,o-wx fich1 fich2
```

 Concede permissões de leitura e execução ao dono e ao grupo e retira permissões de escrita e execução aos outros utilizadores sobre os ficheiros `fich1` e `fich2`;

Nota: Este modelo de definição de permissões não altera as categorias omissas. Por exemplo, no último exemplo nada é referido quanto à permissão de escrita do dono e do grupo nem da permissão de leitura dos restantes utilizadores pelo que estas permissões manter-se-ão como até então. Ao contrário, a utilização do modelo numérico define todas as permissões em simultâneo.

7.8 Informação sobre outros utilizadores do sistema

A lista de todos os utilizadores registados no sistema pode ser obtida por consulta ao ficheiro `/etc/passwd`. Este ficheiro é uma base de dados dos utilizadores onde são mantidos todos os dados relevantes. Cada utilizador tem uma linha com os seguintes dados separados por dois pontos (“:”):

- login do utilizador (account);
- password (cifrada);
- número de identificação do utilizador (UID);
- número de identificação do grupo de utilizador (GID);
- Comentários sobre o utilizador;
- Directoria *HOME*;
- Shell a executar

Todos os utilizadores têm permissão de leitura a este ficheiro.

Outras informações podem ser obtidas através do comando `finger` que fornece dados variados (última vez que recebeu e viu correio electrónico, data da última entrada no sistema, etc.)⁷ sobre um utilizador (eventualmente de outro sistema) indicado como parâmetro. O nome especificado tanto pode ser o *login* como qualquer palavra que se encontre em `/etc/passwd` na zona de comentários. Se existir um ficheiro chamado `.plan` na *HOME* de um utilizador, o seu conteúdo será apresentado como parte dos dados de `finger`.

7.8.1 Utilizadores activos

Num dado instante, é natural que vários utilizadores acedam à mesma máquina, alguns deles eventualmente através de terminais remotos. Para determinar quais os utilizadores que estão activos num dado momento existem os comandos `users`, `who` e `w` que fornecem informação mais detalhada de acordo com esta sequência.

O comando `who am i` auxilia a resolução de “crises de identidade”.

⁷Razões de segurança têm levado a restrições no uso deste comando e à adulteração intencional dos dados por ele fornecidos

7.9 Comandos de processos

Cada utilizador de um sistema Unix tem associados vários processos. Define-se como processo um programa em execução. Sempre que a nível da shell se invoca um comando, são criados um ou mais processos no sistema, para a sua execução.

O Unix dispõe de um comando `ps` (*process status*) que permite determinar quais os processos activos no sistema num dado instante, juntamente com alguma informação adicional, dependente das opções utilizadas. A execução de `ps` sem opções apresenta os processos associados ao utilizador em causa.

7.9.1 Execução de comandos em *background*

Alguns programas não requerem a intervenção do utilizador enquanto são executados. São exemplos destes a compilação de programas e os servidores. Outra característica destes programas é a sua longa duração de execução (no caso dos servidores a sua duração é, supostamente, infinita).

Para evitar que estes programas ocupem indefinidamente uma shell (e portanto uma sessão) enquanto o utilizador pode estar a realizar outras acções, o sistema oferece a possibilidade de executar comandos em *background*.

Um programa em *background* é um programa que é lançado a partir da shell mas em que esta não aguarda pela sua terminação para receber outros comandos. Estes processos não podem receber informação do teclado.

O lançamento de um processo para *background* é feito terminando a linha de comando com `&`. A shell indicará o pid do processo lançado e um identificador interno entre `[]` e apresentará novamente o prompt.

A execução deste processo é indicada pelo comando `ps`.

Se e quando o processo terminar, a shell indicará esse facto ao utilizador afixando no écran a seguinte informação:

```
[id]+ Done <comando>
```

7.9.2 Matar um processo

É frequente acontecer que a execução de um programa dê origem a uma situação da qual parece não haver saída pelos meios mais pacíficos. O comando `kill` para além de enviar sinais para programas permite também terminar processos.

Para tal, basta saber qual o *pid* (process id) do programa em causa, o que pode ser determinado pelo comando `ps`.⁸ A opção para matar um processo é `-9`. Portanto, para matar um dado processo basta executar `kill -9 <pid>`. Note-se que este comando *mata* qualquer processo, em particular, se for dado o pid adequado pode matar a instância da shell em utilização.

Nota: Um utilizador só pode matar processos que tenham sido lançados por si próprio. O administrador do sistema pode realizar esta operação sobre qualquer processo em execução.

⁸Se o programa bloqueou a shell, em Linux é sempre possível abrir uma nova shell (mudando de sessão) e executar `ps` na recém criada.

7.10 Restrições de espaço

A administração do sistema pode impor limites ao espaço ocupado por cada utilizador na sua área. Na gíria do Unix chama-se *quota* ao espaço disponível para cada utilizador. A quota de um utilizador, bem como os recursos consumidos, pode ser vista pelo comando `quota` que produz o seguinte resultado para um utilizador *bem comportado*:

```
Disk quotas for user xpto (uid 1344):  
Filesystem blocks  quota  limit  grace  files  quota  limit  grace  
/dev/hda4   21    5000  5100           7    1000  1100
```

Os valores máximos estão indicados nos campos *quota*.

Nota: Quando é utilizado um sistema de ficheiros distribuído, o comando `quota` utilizado é o da máquina local e não o da máquina remota o que dá uma noção incorrecta dos limites realmente impostos ou até a informação de que o utilizador não sofre de qualquer limitação. Para obter dados correctos será necessário que o utilizador entre na máquina (por exemplo por `ssh` e que será descrito mais adiante neste documento).

Parte III

O Sistema de Janelas X Windows

Capítulo 8

Apresentação do X Windows

O X Windows é um ambiente gráfico para execução de aplicações em sistemas operativos da família Unix. Para além das aplicações explicitamente criadas para este ambiente, o X permite também a execução de aplicações em modo de texto através da abertura de “shells”.

Um dos pontos fortes do X é a possibilidade de abertura de janelas para aplicações a serem executadas noutras máquinas. Isto é, se o utilizador abrir uma shell numa máquina remota (por exemplo por telnet ou ssh) e invocar uma aplicação que use este ambiente gráfico, o servidor local de X oferece todas as facilidades (inclusivé utilização do rato) que estariam disponíveis se a aplicação fosse executada localmente.

Capítulo 9

Entrada e saída de uma sessão de X

Os requisitos necessários para o início de uma sessão de X são os mesmos que para a entrada numa sessão usual de Linux, nomeadamente, o login e a password. Recordamos que o X Windows é executado normalmente na sessão 7 do Linux e portanto, para obter o écran inicial deve ser premido Alt+F7.

Tal como em modo de texto, o sistema afixa o nome do utilizador (login) enquanto este é digitado na janela e não apresenta a password. A mudança entre estes campos é também feita pela tecla Enter.

Uma vez mais, a digitação incorrecta de qualquer um dos elementos fará surgir a mensagem **Login incorrect**.

O abandono da sessão é feito com recurso a uma opção do menu (veremos mais adiante como este pode ser visualizado) ou por `^backspace` que reinicializa o sistema de janelas. O abandono da sessão só deve ser executado depois de encerradas todas as janelas de aplicação e de shells.

Capítulo 10

Elementos básicos de uma sessão de X

O X é altamente configurável e personalizável. Utiliza alguns ficheiros iniciados por "." na HOME do utilizador para obter a configuração pretendida. Como já foi referido, graças ao sistema de ficheiros distribuído, a área de cada utilizador está disponível em qualquer máquina dos laboratórios pelo que a configuração do X será idêntica em qualquer máquina em que seja feito o login.

10.1 Écrans virtuais

O X apoia-se num sistema de écrans virtuais, disponíveis de duas formas, não mutuamente exclusivas.

Um écran pode ter uma dimensão superior à visível no monitor. Quer isto dizer que as janelas podem-se encontrar em zonas não visíveis, que se poderão tornar visíveis deslocando o rato. O scroll é automático pelo deslocamento do rato, ou seja, a zona visível desloca-se de forma a que o rato se encontre sempre sobre ela (figura 10.1).

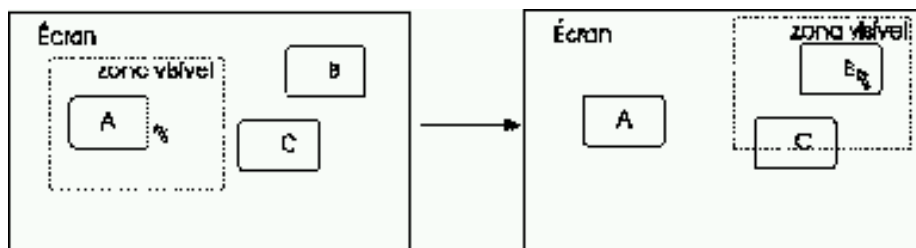


Figura 10.1: Écran de dimensões superiores ao monitor

Um écran pode ser composto por várias páginas, cada uma com o tamanho mínimo da zona visível. Cada página é independente das restantes, contendo as suas próprias janelas abertas. Geralmente, este modelo apresenta um mapa de páginas no canto superior esquerdo do monitor. O deslocamento entre páginas

é feito seleccionando com o rato a página pretendida no mapa ou por uma das combinações $\hat{\leftarrow}$, $\hat{\rightarrow}$, $\hat{\uparrow}$, ou $\hat{\downarrow}$ que muda a página visível no sentido da seta. Encostar o rato à fronteira entre duas páginas provoca também a mudança para a página não visível da fronteira (figura 10.2).

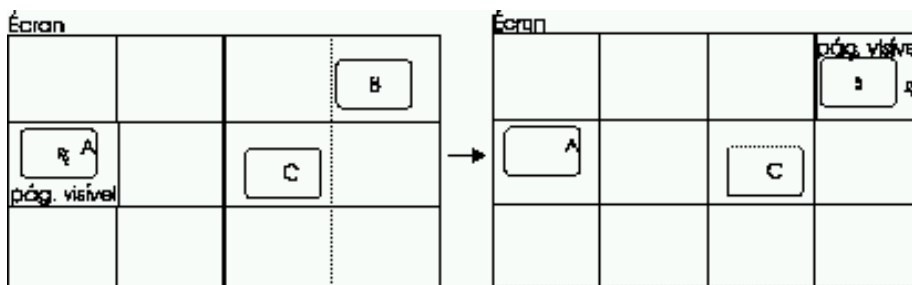


Figura 10.2: Écran de dimensões superiores ao monitor paginado

Combinações destes dois modos são também permitidas. Um écran pode conter várias páginas, sendo cada página de dimensões superiores à zona visível.

10.2 Rato

A utilização do rato no X Windows é semelhante à dos sistemas de janelas da Microsoft. A principal diferença é a utilização de três botões ao contrário dos dois usuais.

Uma vez que o interface do Linux para estes dispositivos assume que os ratos dispõem apenas de 2 botões, mesmo em ratos com 3 botões, o terceiro é emulado primindo os dois outros. A pressão pode ser simultânea ou primindo inicialmente o botão esquerdo e, sem levantar este, primir o botão direito. Note-se que se esta operação for realizada pela ordem inversa, o efeito produzido já não é equivalente.

Em algumas aplicações os botões são identificados pelos números 1, 2 e 3 que correspondem respectivamente aos botões esquerdo, direito e central.

10.3 Menus

Toda a informação de aplicações é apresentada em janelas, que, à semelhança do Windows podem ser deslocadas, minimizadas e maximizadas. Os diálogos com o sistema são feitos por menus, obtidos primindo os botões do rato sobre uma zona livre do écran.¹ O menu de aplicações agrupa normalmente as opções por categorias, devidamente assinaladas por uma seta após o título. Parando o movimento do rato sobre uma categoria faz surgir as opções associadas. O levantar do botão esquerdo do rato selecciona a opção indicada pelo cursor.

Um outro botão (também quando utilizado sobre uma zona livre do écran) faz surgir um menu com a lista das janelas presentemente abertas. Seleccionando uma (libertando o botão) torna-a activa e visível.

¹No gestor de janelas que simula o ambiente do Microsoft Windows 95 o menu pode também ser obtido primindo o botão Start que surge no fundo do écran.

10.4 Utilização

10.4.1 Shells

Uma das opções dos menus é a abertura de shells. Cada shell será aberta numa janela diferente e será independente das restantes. As shells oferecem a possibilidade de utilizar diversas sessões em simultâneo numa mesma máquina com uma interface mais agradável que a proporcionada pela configuração usual do Linux em modo de texto.

As shells podem ser abertas com diferentes configurações de dimensão de fonte e cor de fundo (normalmente preto ou branco com as letras na cor inversa). Cabe ao utilizador seleccionar a da sua preferência.

10.4.2 Invocação de aplicações

A abertura de aplicações pode ser feita directamente a partir do menu ou invocando o executável a partir de uma shell. A aplicação detectará se o utilizador a está a invocar a partir de uma shell aberta em X Windows e será iniciada nesse ambiente.

Para evitar que a shell fique “bloqueada” durante a execução da aplicação, o utilizador poderá lançar a aplicação em background através do procedimento descrito na secção 7.9.1 (página 38).

Aplicações usuais

Algumas das aplicações mais utilizadas neste ambiente são:

emacs/xemacs Editor de texto muito popular e dinâmico que suporta também diversas outras funcionalidades como a gestão de correio electrónico.

ghostview Utilitário de visualização de documentos nos formatos *postscript* e *pdf*.

Netscape Navigator *Browser* da Internet com todas as funcionalidades usuais neste tipo de aplicações.

10.4.3 Abertura de sessões remotas

Uma das propriedades mais relevantes do X Windows é a sua capacidade de suportar o lançamento de janelas abertas em sessões em máquinas remotas.

Ou seja, estabelecida numa shell em X Windows uma ligação a uma máquina remota por intermédio por exemplo do *ssh*,² é possível invocar uma aplicação para este ambiente e utilizar localmente a sua janela de forma transparente. Note-se que a aplicação está a ser executada na máquina remota e portanto toda a informação (por exemplo a árvore de directorias) corresponderá à existente na outra máquina. Este conceito é apresentado na figura 10.3.

Nota: Eventualmente poderá ser necessário instruir o servidor local de X a aceitar pedidos da máquina remota através do comando *xhost* e configurar a máquina remota com a variável de ambiente *DISPLAY*.

²Ver secção 14.1 na página 63.

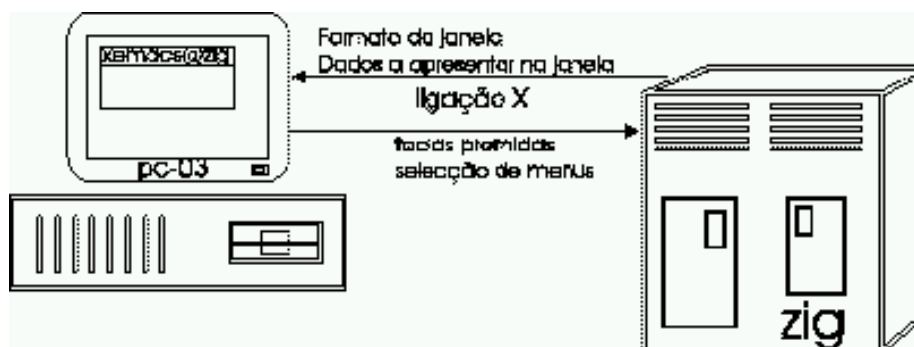


Figura 10.3: Execução de aplicações em máquinas remotas no X Windows

10.4.4 Cópia de dados

O mecanismo de *Copy/Paste* (copiar e colar) do X Windows é mais simples que o seu equivalente nos sistemas operativos Microsoft.

O texto é seleccionado arrastando o rato sobre ele e mantendo o botão esquerdo do rato premido. A operação de *colar* é feita premindo os botões esquerdo e direito do rato em simultâneo.³ O texto entretanto seleccionado será colocado como se tivesse sido digitado na janela activa nesse momento.

Notas:

1. O X Windows não fornece a operação de corte (*cut*).
2. Este mecanismo aplica-se exclusivamente a texto. Para outros tipos de dados (por exemplo imagens) ou operações (por exemplo, o corte) é da responsabilidade das aplicações fornecerem mecanismos equivalentes.

³Que na linguagem da documentação do X Windows é equivalente ao botão 3 do rato.

Capítulo 11

Gestores de janelas

Gestores de janelas são as aplicações que gerem o interface com o utilizador do X Windows. Numa só máquina podem estar simultaneamente em execução vários gestores de janelas, um por cada utilizador.

Do ponto de vista do utilizador, os gestores de janelas diferem no aspecto das janelas (posição e símbolos dos menus, formato dos botões, margens), nos tipos de ícones, nos modelos de écrans virtuais apresentados e no modo de interacção entre o utilizador e o X.

São exemplos de gestores de janelas o fvwm, o twm, o olvwm, o olwm, fvwm2, fvwm2-95¹, AfterStep, AnotherLevel, TheNextLevel, CDE, KDE e WindowMaker.

11.1 Ficheiros relevantes

Os gestores de janelas são personalizados por ficheiros de configuração na HOME do utilizador. Normalmente existem versões *por omissão* na directoria onde se encontra o executável e que podem ser usados como modelo para a personalização.

Uma das diferenças de utilização mais relevantes prende-se com a determinação, em cada instante, de qual a janela que recebe o *input* do teclado (janela activa). Existem duas categorias:

- a janela activa é a última que recebeu um clique do rato sobre ela.
- a janela activa é aquela que se encontra sob o cursor do rato.

Através de uma linguagem própria, estes ficheiros permitem, por exemplo:

- Configurar o menu;
- Configurar as cores e imagens do ambiente;
- Configurar as dimensões dos écrans virtuais e das páginas.
- Lançar automaticamente aplicações aquando da entrada no sistema;

¹Este apresentando uma interface semelhante ao Windows 95

- Determinação da janela activa;

À semelhança das restantes aplicações do sistema operativo Unix, estes ficheiros são iniciados por um ponto (“.”). Exemplos destes ficheiros são o `.fvwmrc` e o `.fvwm2rc` respectivamente para os gestores de janelas *Fvwm* e *Fvwm2*.

Parte IV

Utilitários

Capítulo 12

Editores de texto

12.1 Visual Editor

O Visual Editor (`vi`) é o editor de texto *standard* dos sistemas operativos da família Unix. Por ter sido pensado para suportar a edição de texto em qualquer terminal, utiliza para comandos apenas as teclas que têm que figurar obrigatoriamente em todos os teclados, nomeadamente, os caracteres alfanuméricos e a tecla `Ctrl`.

Esta particularidade confere ao `vi` um interface pouco vulgar com o utilizador quando comparado com outras ferramentas de edição de texto. Em contrapartida, é o único editor de texto que será certamente encontrado em qualquer sistema Unix.

O `vi` é invocado por:

```
vi [opcoes] {ficheiro}
```

12.1.1 Modos de edição

O `vi` tem dois modos de operação:

comandos Em que o cursor é livremente deslocado sobre o texto através das teclas `h,j,k,l` (respectivamente esquerda, baixo, cima, direita).¹ É ainda permitido eliminar caracteres (tecla `x`) e executar diversas outras operações como pesquisa, substituição, etc.

inserção Em que é adicionado texto ao écran a partir do ponto em que o cursor se encontra. A eliminação de caracteres é também permitida (`backspace` ou `^h`) bem como a inserção de linhas (tecla `Enter`).

A comutação entre estes dois modos é feita por:

- Do modo de comandos para o modo de inserção por uma das teclas `a`, `A`, `I`, `o`, `O` que oferecem diferentes opções de posicionamento para inserção.²
- Do modo de inserção para o modo de comandos pela tecla `Esc`.

¹Versões mais recentes suportam também as teclas de deslocamento do cursor `←`, `↓`, `↑`, `→` para estas operações.

²Ver anexo C na página 81 para uma descrição detalhada destas funções.

12.1.2 Outras funções

O `vi` dispõe de todas as funcionalidades esperadas de um editor de texto. Operações de cópia e deslocação de texto, pesquisa, substituição, etc. estão também disponíveis.

A saída da aplicação é feita pelo comando `:q<Enter>` em modo de comandos. A gravação do documento em edição pelo comando `:w [nome] <Enter>`.

Os comandos do `vi` estão listados no Anexo C (pág. 81).

12.2 Joe

O `joe` (Joe's Own Editor) é um dos exemplos do modelo de disponibilização de software para o sistema operativo Linux. Apesar de ter sido criado por Joseph H. Allen para seu uso pessoal foi disponibilizado gratuitamente e conseguiu uma grande aceitação entre os utilizadores do Linux.

É um editor de ficheiros para modo de texto que suporta o processamento de diversos ficheiros em simultâneo e operações de edição entre eles. Por configuração podem ser definidas formatações automáticas consoante o tipo de ficheiro em edição (de língua natural, de linguagem de programação, ...).

O `joe` é invocado por:

```
joe [opções] {ficheiro}
```

O écran do `joe` divide-se em três áreas:

- No topo mantém informações de *status* do ficheiro em edição: n^o da linha e coluna em que se encontra o cursor, nome do ficheiro, estado do ficheiro relativamente a alterações e informação genérica (hora e informação de ajuda). A zona superior pode ser alargada de forma a conter páginas de ajuda que apresentam todos os comandos disponíveis.
- No centro encontra-se a área de edição do documento.
- Na parte inferior são mantidos os diálogos com o utilizador.

Quando o teclado se encontra configurado adequadamente, o `joe` suporta as teclas de deslocamento, como as setas de direcção e Page Up e Page Down. Os comandos mais usuais do `joe` estão descritos no apêndice B, na página 79.

Em todos os documentos abertos simultaneamente existe no máximo um bloco definido. Assim, é possível copiar/mover texto entre ficheiros marcando o bloco desejado, mudando para o ficheiro de destino e executando o comando de cópia ou transferência.

12.2.1 Ficheiro de configuração

Quando inicializado, o `joe` procura um ficheiro de configuração com o nome `.joerc` respectivamente nas seguintes directorias:

1. Directoria corrente
2. `HOME` do utilizador
3. Caso não encontre, utiliza o ficheiro `joerc`³ definido na directoria `/usr/local/lib`.

³Que pode ser utilizado como base para a criação do pessoal.

Este ficheiro permite uma grande capacidade de personalização do editor oferecendo, por exemplo, opções de formatação dependentes da extensão do ficheiro, configuração da directoria onde são guardadas as cópias de backup⁴ e configuração das teclas de comandos.

12.3 emacs

O emacs ultrapassa as funções de edição de texto, podendo também ser utilizado para, entre outros, agente de correio electrónico, gestor de ficheiros, etc.

O conceito fundamental deste editor é o de *buffer*. Um buffer é um espaço que contém alguma forma de dados, sejam eles, documentos de texto, mensagens de correio electrónico, listas de ficheiros ou outros.

A interface do emacs é bastante agradável, em particular na sua versão para o X Windows (o xemacs) onde estrutura as suas opções numa barra de menus seleccionáveis pelo rato.

⁴Com o mesmo nome do ficheiro inicial mas terminadas com ~

Capítulo 13

Correio electrónico e newsgroups

O correio electrónico (e-mail) é uma das ferramentas de comunicação oferecidas pela Internet e para o qual existem utilitários de envio/recepção para praticamente todos os sistemas operativos.

Quase todos os *browsers* dispõem actualmente de módulos que permitem a gestão de correio electrónico de uma forma muito agradável.

Uma mensagem de correio electrónico é composta por duas zonas distintas:

Cabeçalho Onde são colocados campos com informações referentes à mensagem, por exemplo:

To Destinatário da mensagem;

Cc (Cópia de carbono) Utilizadores que receberão cópias da mensagem;

Bcc (Blind Carbon Copy) Utilizadores que receberão cópias da mensagem mas dos quais os destinatários indicados em **To:** e **Cc:** não terão conhecimento;

Subject Título da mensagem.

Attachments Outros ficheiros que serão enviados conjuntamente com a mensagem. Actualmente estes ficheiros podem conter qualquer tipo de dados.

Corpo Texto da mensagem.

Os campos **To**, **Cc** e **Bcc** podem receber um ou mais endereços separados por vírgulas (“,”).

Os servidores de mail tentam entregar a mensagem o mais depressa possível à máquina de destino. O prazo de validade de uma mensagem é normalmente de 5 dias. Se neste prazo o servidor não conseguir entregar a mensagem ou um dos endereços especificados não exista, o emissor será notificado por uma mensagem.

13.1 Regras de utilização de newsgroups

Os newsgroups são fóruns de discussão de acesso livre. Existem milhares destes fóruns, organizados por temas. Os newsgroups são normalmente identificados por contextos, separados por pontos finais (“.”). Por exemplo:

informatica.disciplinas.fso Identifica o newsgroup da cadeira fso do departamento de informática.

pt.geral Identifica o newsgroup de âmbito geral de Portugal e onde a discussão é feita em português

Por se tratar de um meio “partilhado”, existem algumas regras a respeitar e que são aqui levemente referidas:

- Antes de enviar uma mensagem para um newsgroup¹ o utilizador deve acompanhar as discussões que nele decorrem durante 15 dias. Desta forma poderá:
 - Observar o tipo de linguagem utilizado nas discussões;
 - Não enviar mensagens com ideias que tenham já sido referidas;
 - Compreender qual o âmbito das discussões que estão a decorrer evitando enviar mensagens fora do contexto do grupo.
- Não deve utilizar linguagem grosseira ou ofensiva para os restantes leitores;
- Respeitar as opiniões manifestadas pelos restantes utilizadores;
- Não enviar mensagens que não se enquadrem no âmbito do newsgroup;

13.2 Pine

O *pine* é um dos agentes de correio electrónico em modo de texto mais utilizados no ambiente Linux.

Envia e apresenta as mensagens de correio electrónico recebidas e gere *newsgroups*. Permite o arquivamento das mensagens em pastas² criadas pelo utilizador. Dispõe de um livro de endereços que pode também ser utilizado como auxiliar na determinação do arquivador a utilizar para as mensagens.

Se for executado numa shell do ambiente X Windows lança automaticamente os utilitários que permitem a visualização de tipos complexos de dados associados às mensagens.

Por omissão, as mensagens recebidas serão colocadas no *folder* Inbox. O utilizador tem a opção de criar outros folders onde pode guardar as mensagens que envie/receba.

¹Esta operação é normalmente denominada de *post*

²*folders*

13.2.1 Écran

O écran do pine é decomposto em três secções:

Topo Onde é apresentada informação de estado, arquivo aberto, mensagem em visualização, ...

Centro Onde são visualizados os dados referentes ao contexto em que o utilizador se encontra

Fundo Onde são apresentadas as opções que o utilizador dispõe no contexto em que se encontra e as teclas associadas. Tal como neste documento, o pine utiliza o símbolo $\hat{}$ para denotar a tecla **Ctrl**.

13.2.2 Configuração

Quando é lançada (pelo comando `pine`) a aplicação apresenta um menu de onde se pode seleccionar (através da tecla **S** ou usando as teclas de direcção do cursor e premindo `<Enter>`) a opção de configuração.

Esta opção dá acesso a um submenu que surge na parte inferior do écran e onde se pode:

- Configurar a impressora (opção `Printer`);
- Configurar o pine (opção `Config`);
- Definir/alterar a assinatura³ (opção `Signature`);
- Definir uma nova password para acesso ao sistema (opção `Newpassword`);
- Actualizar a versão do Pine (opção `Update`);

A opção `Config` é aquela que é mais relevante para a utilização do Pine. A edição das opções de configuração é feita deslocando a barra de selecção com as teclas de direcção e premindo `Enter`. O pine dispõe de ajuda para todas as opções apresentadas através do comando `“?”`.

É nesta opção que são configurados os servidores (de newsgroups e eventualmente de correio electrónico) a utilizar, a identificação do utilizador, o tempo entre verificações de recepção de novas mensagens, etc.

³A assinatura é um bloco de texto que é incluído automaticamente no fim de todas as mensagens enviadas

Capítulo 14

Login remoto

14.1 Secure Shell (ssh)

O Secure Shell permite o estabelecimento de uma comunicação segura ao cifrar todas as mensagens enviadas nas duas direcções. A cifra é também aplicada a conexões criadas pelo ambiente gráfico X Windows para execução remota de aplicações.

O formato de invocação é:

```
ssh [opcoes] host [comando]
```

Se a opção comando não for indicada, o ssh apresenta uma shell no terminal remoto. Se não for especificado um username (através da opção -l) o ssh procederá a uma tentativa de login com o nome com que o utilizador está presentemente registado na máquina local.

As vantagens de utilização do ssh sobre o telnet são evidentes: toda a comunicação trocada entre as duas máquinas é cifrada, inclusive a password, o que inutiliza tentativas de ataque por escuta da rede. Actualmente, a política do departamento é a de restringir sempre que possível a utilização do telnet, havendo mesmo alguns servidores que aceitam apenas ligações seguras.

14.1.1 Problemas de autenticação

O ssh conserva as chaves apresentadas pelos servidores com quem foi contactado. Se o servidor, por alguma razão (por exemplo, um crash do disco) mudar de chave, o ssh detecta-o e considera a hipótese de estar a decorrer um ataque à máquina. Para manter a segurança do utilizador não realiza a operação e apresenta a mensagem:

```
@ WARNING: HOST IDENTIFICATION HAS CHANGED! @
#####
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the host key has just been changed.
Please contact your system administrator.
Add correct host key in /users/xpto/.ssh/known_hosts to get rid of this message.
Password authentication is disabled to avoid trojan horses.
Permission denied.
```

Após se assegurar que não está de facto a ocorrer um ataque (por exemplo, se for público que o sistema operativo foi reinstalado no servidor) o utilizador deverá editar o ficheiro `.ssh/known_hosts` e eliminar a linha referente ao servidor em causa. Quando for pedida uma nova ligação, o ssh indicará que está a contactar um servidor desconhecido e adicionará a nova chave ao ficheiro.

Capítulo 15

Arquivo e compressão

15.1 Tape Archive Utility (tar)

O tar é um programa de arquivo de ficheiros, ou seja, agrupa diversos ficheiros num único, normalmente com extensão `.tar`. O seu propósito inicial era facilitar as operações de backup para unidades de banda. No entanto, a sua versatilidade fez com que fosse utilizado sempre que é vantajoso agrupar ficheiros relacionados. É o formato mais utilizado para downloads de aplicações para os sistemas operativos da família Unix.

O tar tem o seguinte formato de invocação:

```
tar [OPTION] ... [FILE] ...
```

O conjunto de opções que se apresenta é apenas uma pequena lista da totalidade.

A primeira opção indica a operação desejada. As mais usuais são:

x eXtrair um ou mais ficheiros do arquivo

c Criar um arquivo

a Adicionar um ficheiro a um arquivo

t Testar e listar o conteúdo de um arquivo

Outras opções usuais são:

z que indica que a operação a realizar deverá ter em conta a compressão do arquivo. Nesse caso os arquivos são identificados por uma das extensões `.tar.gz` ou `.tgz`

f **ficheiro** nome do arquivo a criar ou a utilizar. Se esta opção não for indicada o tar assume que o arquivo se encontra no dispositivo de *backup* da máquina. Dado como ficheiro o caracter '-' redirecciona o input ou o output para o stdin ou stdout (consoante o tipo de operação) permitindo a utilização de pipes.

Os ficheiros indicados em seguida são os ficheiros a arquivar/extrair ao arquivo. No caso da extracção, se não for indicado qualquer ficheiro, o programa fará a extracção de todos os ficheiros contidos pelo arquivo. Em qualquer um dos casos podem ser indicadas directorias, assumindo-se, por omissão, que se pretende a criação/extracção de todos os ficheiros e subdirectorias nela contidas.

15.1.1 Exemplos de utilização

`tar xzf backup.tar.gz` extrai todos os ficheiros do arquivo compactado `backup.tar.gz` que se encontra na directoria corrente.

`tar xzf back.tgz fich1.txt` extrai o ficheiro `fich1.txt` do arquivo compactado `back.tgz` que se encontra na directoria corrente.

`tar cf back.tar f1.txt mydir` cria um arquivo com o nome `back.tar` na directoria corrente contendo o ficheiro `f1.txt` e todo o conteúdo da directoria `mydir`.

Notas:

1. Algumas versões mais antigas do `tar` (não GNU) não suportam a utilização da opção `z`. Nesse caso a extracção de ficheiros de arquivos compactados poderá ser feita utilizando inicialmente o comando `gzip` para proceder à descompressão e em seguida aplicando o `tar`. Através de pipes a operação poderá ser realizada directamente. Por exemplo: `gzip -dc backup.tar.gz | tar xf - fich1.txt` extrai o ficheiro `fich1.txt` do arquivo `backup.tar.gz`.
2. As versões mais recentes do `winzip` (para Windows) suportam ficheiros de arquivo gerados por `tar`, compactados ou não com `gzip`.
3. O `tar` é um comando muito rico no conjunto de opções que oferece pelo que se aconselha a leitura da página de manual correspondente.

15.2 GNU Zip

O `gzip` é um utilitário de compressão de ficheiros. Programas de compressão são aplicações que reduzem a dimensão dos ficheiros através de algoritmos que eliminam a redundância dos dados.¹ Consoante o tipo de dados, a aplicação que os gerou e a dimensão do ficheiro, estes utilitários chegam a reduzir os dados guardados a 30% ou menos da dimensão inicial. Estes programas asseguram que quando novamente descompactados, os dados obtidos serão precisamente os mesmos.

O único senão destes algoritmos é a impossibilidade de as aplicações que geraram os ficheiros os “entenderem” enquanto compactados e o tempo consumido por estas operações. Por essas razões, estas aplicações são utilizadas apenas para arquivo de ficheiros.

O `gzip` tem o seguinte formato de invocação:

```
gzip [opcoes] ficheiro1 {ficheiro2}
```

Quando invocado sem opções, o `gzip` comprime o(s) ficheiro(s) indicado(s) concatenando ao nome a extensão `.gz`. Os ficheiros só podem ser compactados uma única vez o que é detectado pelo `gzip` através da análise à extensão do ficheiro.

A descompressão é feita pela opção `-d`. Nesse caso, ao ficheiro resultante será automaticamente extraída a extensão `.gz`.

A opção `-c` redirecciona o resultado de qualquer uma das operações para o standard output permitindo assim a utilização de redireccionamento e pipes.

¹Vulgarmente designados por algoritmos de compressão.

15.3 Zip

O Zip é um utilitário de compressão e arquivo. Ou seja, permite a compressão de diversos ficheiros num único.

Este comando tem um conjunto de opções muito rica, que permite uma selecção de ficheiros muito eficiente. Os ficheiros criados com este utilitário adquirem por omissão a extensão “.zip” e podem ser compactados/descompactados pelo utilitário `wzip` para Windows.

A função de descompressão é realizada pelo comando `unzip`.

15.4 Compress

O comando `compress` é o utilitário *standard* de compressão de ficheiros do Unix. A extensão por omissão dos ficheiros gerados é “.Z”. Este comando é menos rico que `zip` na selecção dos ficheiros a arquivar.

O comando `uncompress` realiza a descompressão de arquivos.

Capítulo 16

Cópia de Ficheiros

16.1 File Transfer Protocol

O ftp é o protocolo de cópia de ficheiros entre servidores da Internet. As ferramentas clássicas apresentam um interface de linha de comando que pode ser “mascarado” pelo acesso através de *browsers*¹ que listam o conteúdo das directorias e permite a utilização do rato para as operações de *download*.

Uma sessão de ftp é iniciada pelo comando

```
ftp [host]
```

onde host é o nome da máquina remota num dos formatos usuais da rede Internet.²

Após o estabelecimento da ligação ao servidor remoto e verificada a sua disponibilidade é validada a identidade do utilizador, a qual poderá ser feita em dois modos:

anónimo: utilizando um dos *usernames*: *Anonymous*, *ftp* ou *guest* que darão acesso aos ficheiros disponibilizados publicamente pelo servidor. É este o modo utilizado pelos *browsers*. No campo de *password* será solicitado ao utilizador a digitação do seu endereço de correio electrónico (e-mail). Na maioria dos servidores este campo é opcional podendo ser ignorado digitando <Enter>.

peçoal: utilizando o *username* de um utilizador registado no sistema remoto o que dará acesso a todos os ficheiros da sua área pessoal. Neste caso, o campo password deverá conter a sua password para o sistema.³

Em qualquer um dos casos a password não será ecoada no écran. Caso a validação seja recusada, o comando **user** permite reiniciar o processo.

Os utilizadores em modo anónimo estão normalmente sujeitos a restrições severas de operação no servidor remoto. Num sistema Unix, as permissões concedidas são as de *other*, não lhes sendo, em princípio concedida a possibilidade de escrever nas directorias ou nos ficheiros.

¹O acesso a servidores ftp a partir de *browsers* é feito substituindo a indicação do protocolo http:// pela indicação do protocolo ftp://

²Por exemplo: caravela.di.fc.ul.pt ou 194.117.20.5

³Chama-se a atenção que a *password* e os dados circulam em “claro” na rede possibilitando a sua obtenção ilegítima por terceiros. Este problema foi resolvido pelo comando scp apresentado na secção seguinte.

A partir do momento em que o utilizador está validado, é aberta uma sessão que será terminada com o comando `bye`.

Os comandos disponíveis estão resumidos na tabela 16.1

<code>?</code>	Lista no écran os comandos disponíveis.
<code>open <host></code>	Estabelece uma ligação com <code>host</code> .
<code>close</code>	Encerra a sessão no servidor remoto.
<code>bye</code>	Termina a aplicação.
<code>lcd <dir></code>	Muda a directoria corrente local.
<code>ls <dir></code>	Lista o conteúdo da directoria remota.
<code>cd <dir></code>	Muda a directoria corrente da sessão remota. A sua sintaxe é semelhante à do comando <code>cd</code> do Unix.
<code>get <ficheiro></code>	Importa <code><ficheiro></code> da máquina remota, colocando-o com o mesmo nome na directoria corrente local.
<code>mget <ficheiro></code>	Versão de <code>get</code> que permite a utilização de <i>wildcards</i> (* e ?).
<code>put <ficheiro></code>	Coloca na directoria corrente da sessão remota o ficheiro indicado.
<code>mput <ficheiro></code>	Versão de <code>put</code> que permite a utilização de <i>wildcards</i>

Tabela 16.1: Resumo dos comandos do programa ftp

16.1.1 Modo de transferência

Nos sistemas operativos da Microsoft, o fim da linha de um ficheiro de texto é assinalado pela sequência de caracteres ASCII de valor decimal 10 e 13 (LF e CR). Nos sistemas Unix é utilizado apenas o carácter LF.

O ftp dispõe de um modo de conversão automática que adapta os ficheiros ao sistema de destino. Para que o ftp realize essa conversão deverá ser colocado em modo ASCII digitando o comando `ascii` (ou `type ascii`). O comando que realiza as transferências sem conversão é `binary` (ou `type binary`).

Note-se que esta conversão nem sempre é desejável, devendo apenas ser aplicada a ficheiros de texto. Ficheiros de outras aplicações (ou até mesmo executáveis) não deverão ser convertidos, ficando muito provavelmente inutilizados.

Chama-se por isso a atenção que a aplicação ftp para Windows é, por omissão, iniciada em modo `ascii`.

16.2 Secure Copy (scp)

O Secure Copy (`scp`) é um programa de transferência de ficheiros dos mesmos autores do Secure Shell (`ssh`). O seu objectivo é garantir a importação/exportação de ficheiros para uma máquina remota de forma segura, ou seja, cifrando a password e os dados na sua passagem pela rede. Ao contrário do `ftp`, o `scp` não concretiza o conceito de sessão sendo os ficheiros a transferir descritos na linha de comando. O formato do comando é:

```
scp [-pr] fichOrig1 {fichOrig2} fichDest
```

Se for indicado mais de um ficheiro de origem o destino deverá ser uma directoria. Apenas os ficheiros origem ou o ficheiro/directoria de destino poderão ser remotos mas não ambos. Os ficheiros remotos são definidos na forma:

`utilizador@host:NomeDoFicheiro`

onde o nome do ficheiro é um *pathname* segundo as regras especificadas na secção 5.8 (pág. 20). A directoria corrente considerada é a *HOME* do utilizador.

É válida a utilização de *wildcards* e do carácter '.' para indicação da directoria corrente. A opção `-r` copia recursivamente directorias.

Após o estabelecimento da ligação ao servidor remoto, será pedida a password do utilizador.

Notas:

1. O modo de autenticação dos servidores remotos e a cifra é partilhada pelo `scp` e pelo `ssh`. Será por isso possível o surgimento da mensagem de erro descrita na secção 14.1.1 (pág. 63). O procedimento para retomar a ligação é semelhante ao então descrito.
2. A inexistência de uma sessão obriga à execução de múltiplos comandos para a importação/exportação de conjuntos de ficheiros. Sugere-se por isso que os ficheiros sejam previamente arquivados por exemplo com recurso ao comando `tar` (pág. 65).
3. A necessidade de cifrar/decifrar a informação acarreta algum atraso na transferência dos ficheiros. Sugere-se por isso que ficheiros de grande dimensão sejam compactados por exemplo com recurso ao comando `gzip` (pág. 66).

16.2.1 Exemplos de utilização

`scp fso000@zig.di.fc.ul.pt:mydir/fich1.txt .` Importa para a máquina local o ficheiro `fich1.txt` da directoria `mydir` do utilizador `fso000` na máquina `zig`.

`scp prog.c fso000@zig.di.fc.ul.pt:prog1dir` Coloca na directoria `prog1dir` que se encontra imediatamente abaixo da *HOME* de `fso000` o ficheiro `prog.c` que se encontra na directoria corrente da máquina local.

16.3 Acesso a drives de diskettes

As estações de trabalho dos sistemas Unix não estavam preparadas para suportar drives de diskettes. Uma vez que o Linux concretiza uma filosofia em que a “estação de trabalho” é um computador pessoal, faz sentido fornecer ferramentas que permitam a utilização do(s) seu(s) drive(s). Estes comandos, que no seu conjunto são denominados *mtools* permitem a execução de todas as operações básicas sobre diskettes.

O formato utilizado é compatível com o MS-DOS o que possibilita a transferência de ficheiros entre os dois sistemas.

A tabela 16.2 apresenta alguns dos comandos da família *mtools*. O nome dos comandos é iniciado por um `m` seguido do nome equivalente no MS-DOS.

A indicação do drive é feita respeitando a sintaxe do MS-DOS (A:,B:), no entanto, as directorias deverão ser especificadas através da barra de divisão ('/').

Note-se que por os comandos operarem sobre drives de diskettes, a sua execução em shells remotas não faz sentido por nesse caso estar a ser feita referência

<code>mcd</code>	Mudança da directoria corrente.
<code>mcopy</code>	Cópia de ficheiros de/para discos MS-DOS.
<code>mdel</code>	Eliminação de ficheiros.
<code>mdeltree</code>	Eliminação de árvores de directorias.
<code>mdir</code>	Listagem do conteúdo de uma directoria.
<code>mformat</code>	Formatação de um disco em formato MS-DOS.
<code>mmd</code>	Cria uma directoria.
<code>mmove</code>	Move ficheiros ou subdirectorias.
<code>mrd</code>	Elimina directorias.
<code>mren</code>	Renomeação de ficheiros.
<code>mtype</code>	Mostra o conteúdo de um ficheiro.

Tabela 16.2: Alguns comandos da família *mtools*

ao drive de diskette remoto. Nesses casos, o ficheiro deverá primeiro ser importado para a máquina local (por scp ou ftp) e só depois copiado para o drive. Em sistemas de ficheiros distribuídos,⁴ mantém-se o princípio da transparência de localização. Assim, os ficheiros poderão ser copiados para o PC e o sistema encarrega-se de os colocar no servidor.

Nota: O comando `mcd` guarda a directoria corrente da drive num ficheiro de configuração da *HOME* do utilizador. Operações subsequentes geram consultas a esse ficheiro. Se o utilizador trocar a diskette do drive com uma outra que não contenha a directoria corrente da anterior, os comandos são incapazes de detectar esse problema e retornam erros. A solução passa por uma de três possibilidades:

1. Retirar diskettes do drive apenas depois de ter executado o comando `mcd a:/`.
2. Repor a diskette no drive, efectuar a operação descrita no ponto anterior e retirar a diskette.
3. Utilizar sempre comandos com caminhos absolutos (a partir da raíz, por exemplo, `mdir a:/`).
4. Eliminar o ficheiro de configuração `.mcd` da *HOME* do utilizador.

16.3.1 Exemplos de utilização

`mcopy projectos/fase1/myfich.txt a:/drivedir1/drivedir2` Copia o ficheiro `projectos/fase1/myfich.txt` para a directoria `drivedir1/drivedir2` da diskette.

`mcd a:/dir2` Torna `dir2` a directoria corrente para acesso ao drive.

`mmove a:fase1/myfich.txt .` Copia o ficheiro `myfich.txt` da directoria `fase1` que se encontra abaixo da directoria corrente para o drive de diskettes para a directoria corrente na sessão de Unix.

⁴Como é o caso do servidor de alunos do departamento (caravela)

16.4 Split

O comando `split` parte um ficheiro em fracções de dimensão configurável. É útil para o transporte de ficheiros de grandes dimensões (por exemplo, arquivos ou *backups*) em dispositivos de pequena dimensão como as diskettes.

O `split` tem a seguinte sintaxe:

```
split [opcoes] [ficheiro [prefixo]]
```

A opção mais útil é a opção `-b` que permite a definição da dimensão máxima dos blocos. O valor é complementado por um indicador de unidade que pode ser *b* (blocos de 512 bytes), *k* (kilobytes) ou *m* (megabytes). O valor por omissão é 1000 linhas (opção `-l`).

O nome de cada bloco será a concatenação de prefixo com uma sequência que se inicia em *aa* e termina em *zz* (o segundo bloco tem o valor *ab*). Se `prefixo` não for indicado, é assumido o prefixo *x*.

Não existe um comando que faça a concatenação dos blocos. Esta poderá ser feita através de redireccionamento e, por exemplo, do comando `cat`.

16.4.1 Exemplos de utilização

```
split -b 1400k fichlarge Divide o ficheiro fichlarge em blocos de 1400 kilobytes. Os blocos terão o nome xaa, xab, xac, ...
```

```
split fichlarge fich Divide o ficheiro fichlarge em blocos de 1000 linhas. Os blocos terão o nome fichaa, fichab, fichac, ...
```

```
cat x* > fichlarge Concatena os blocos do primeiro exemplo num único ficheiro chamado fichlarge. A ordenação dos blocos está garantida por o metacaracter * ordenar alfabeticamente os ficheiros.
```


Apêndice A

Resumo dos comandos do GNU Emacs

GNU Emacs Reference Card

(continued)

Starting Emacs

To start GNU Emacs, type `emacs` in a shell, or `emacs &` to start it in the background.

Leaving Emacs

To quit Emacs, type `q` or `Ctrl-C`.

Files

`Ctrl-F` read a file into Emacs.
`Ctrl-R` save a file back to disk.
`Ctrl-S` save all files.
`Ctrl-W` save all files.
`Ctrl-X` save all files.
`Ctrl-Z` save all files.
`Ctrl-^` save all files.
`Ctrl-~` save all files.

Getting Help

The help system is available. Type `Ctrl-h` to read the help files.
`Ctrl-h` show the help files.
`Ctrl-h` show the help files.
`Ctrl-h` show the help files.
`Ctrl-h` show the help files.
`Ctrl-h` show the help files.

Error Recovery

Most errors are recoverable. Type `Ctrl-y` to redo the last command.
`Ctrl-y` redo the last command.
`Ctrl-y` redo the last command.
`Ctrl-y` redo the last command.
`Ctrl-y` redo the last command.

Incremental Search

`Ctrl-s` start incremental search.
`Ctrl-s` start incremental search.
`Ctrl-s` start incremental search.
`Ctrl-s` start incremental search.
`Ctrl-s` start incremental search.
`Ctrl-s` start incremental search.
`Ctrl-s` start incremental search.
`Ctrl-s` start incremental search.
`Ctrl-s` start incremental search.
`Ctrl-s` start incremental search.

Motion

`Ctrl-L` end of line.
`Ctrl-H` beginning of line.
`Ctrl-A` beginning of buffer.
`Ctrl-E` end of buffer.
`Ctrl-B` backward one character.
`Ctrl-F` forward one character.
`Ctrl-N` next line.
`Ctrl-P` previous line.
`Ctrl-V` copy text.
`Ctrl-W` kill text.
`Ctrl-X` undo.
`Ctrl-Z` redo.
`Ctrl-^` undo.
`Ctrl-~` redo.

Killing and Deletion

`Ctrl-K` kill text.
`Ctrl-D` delete text.
`Ctrl-^` undo.
`Ctrl-~` redo.
`Ctrl-W` kill text.
`Ctrl-X` undo.
`Ctrl-Z` redo.
`Ctrl-^` undo.
`Ctrl-~` redo.

Marking

`Ctrl-M` mark the beginning of the line.
`Ctrl-@` mark the beginning of the line.
`Ctrl-@` mark the beginning of the line.
`Ctrl-@` mark the beginning of the line.
`Ctrl-@` mark the beginning of the line.

Query Replace

`Ctrl-R` query replace.
`Ctrl-R` query replace.
`Ctrl-R` query replace.
`Ctrl-R` query replace.
`Ctrl-R` query replace.

Multiple Windows

`Ctrl-W` toggle between windows.
`Ctrl-W` toggle between windows.
`Ctrl-W` toggle between windows.
`Ctrl-W` toggle between windows.
`Ctrl-W` toggle between windows.
`Ctrl-W` toggle between windows.
`Ctrl-W` toggle between windows.
`Ctrl-W` toggle between windows.
`Ctrl-W` toggle between windows.
`Ctrl-W` toggle between windows.

For mousing

`Ctrl-M` mouse click.
`Ctrl-M` mouse click.
`Ctrl-M` mouse click.
`Ctrl-M` mouse click.
`Ctrl-M` mouse click.
`Ctrl-M` mouse click.
`Ctrl-M` mouse click.
`Ctrl-M` mouse click.
`Ctrl-M` mouse click.
`Ctrl-M` mouse click.

Case Change

`Ctrl-U` uppercase.
`Ctrl-L` lowercase.
`Ctrl-C` capitalize.
`Ctrl-W` toggle case.
`Ctrl-W` toggle case.
`Ctrl-W` toggle case.
`Ctrl-W` toggle case.
`Ctrl-W` toggle case.

The Minibuffer

The minibuffer is used for prompts.
`Ctrl-C` abort command.
`Ctrl-E` end of line.
`Ctrl-N` next line.
`Ctrl-P` previous line.
`Ctrl-W` kill text.
`Ctrl-X` undo.
`Ctrl-Z` redo.
`Ctrl-^` undo.
`Ctrl-~` redo.

Apêndice B

Resumo dos comandos do joe

^K h	Alterna a existência de informação de ajuda na zona superior do écran;
^K e	Abertura de um ficheiro, na linha inferior o utilizador é convidado a indicar o nome do ficheiro pretendido. A tecla TAB produz o mesmo resultado de quando utilizada na shell. A abertura de um ficheiro não provoca a eliminação dos restantes.
^K s	Gravação. O joe propõe a utilização do nome do ficheiro mas é possível a sua alteração. Por omissão, a versão anterior do ficheiro será renomeada acrescentando um til.
^K x	Gravação do ficheiro e abandono. Se for o único ficheiro em utilização o joe termina.
^C	Termina a edição do ficheiro corrente sem gravar as alterações. Se o ficheiro tiver sido alterado pede confirmação.
^K b	Definição do início do bloco. Os blocos de texto permitem as operações de cópia, deslocamento e eliminação.
^K k	Definição de fim de bloco.
^K c	Cópia do bloco para a posição actual do cursor.
^K m	Deslocamento do bloco para a posição actual do cursor.
^K y	Eliminação do bloco.
^K n	Edição do ficheiro seguinte na lista (se algum).
^K p	Edição do ficheiro anterior na lista (se algum).
^T	Opções de configuração, que surgem na janela inferior. Apresenta apenas três opções por linha. A navegação entre opções é feita pelas teclas de deslocamento do cursor ou digitando a letra em maiúsculas da opção. A selecção de uma opção é feita premindo Enter ou a barra de espaços. Entre as opções destacam-se: <i>Autoindent</i> que faz com que quando é criada uma linha nova o cursor se posicione automaticamente sob o primeiro carácter não espaço da linha anterior; <i>Word wrap</i> que gera uma mudança automática de linha sempre que é ultrapassada a margem direita (também definida por uma opção).

Apêndice C

Resumo dos comandos do vi

Comandos de criação

- a Criar texto após o cursor
- A Criar texto no fim da linha corrente
- i Criar texto sob o cursor
- I Criar texto antes do primeiro caracter na linha que não é um caracter em branco
- o Criar texto no início de uma nova linha, abaixo da linha corrente
- O Criar texto no início de uma nova linha, acima da linha corrente
- <ESC> Retornar o vi ao modo de comandos de qualquer um dos modos de entrada de texto acima

Comandos de eliminação

Em modo de inserção	
<code>backspace/ ^ h</code>	Eliminar o caracter corrente
<code>^w</code>	Eliminar a palavra actual
<code>@</code>	Eliminar a linha actual de texto novo ou eliminar todo o texto novo na linha actual
Em modo de comando	
<code>u</code>	Desfazer o último comando
<code>U</code>	Apagar a última alteração na linha corrente
<code>x</code>	Apagar o caracter corrente
<code>ndx</code>	Apagar n objectos de texto x
<code>dw</code>	Apagar a palavra sob o cursor até ao próximo espaço ou marca de pontuação
<code>dd</code>	Apagar a linha corrente
<code>D</code>	Apagar do cursor até ao fim da linha
<code>d)</code>	Apagar a frase corrente
<code>d}</code>	Apagar o parágrafo corrente

Comandos de modificação

- r Substituir apenas o caracter corrente
- R Substituir apenas os caracteres sobrepostos até que seja dado o comando <ESC>
- s Eliminar o caracter sob o cursor e adicionar texto até que seja dado o comando <ESC>
- S Substituir todos os caracteres na linha
- cw Substituir uma palavra ou os caracteres em falta de uma palavra por texto novo
- cc Substituir todos os caracteres da linha
- ncx Substituir n objectos de texto x tais como frases (“)”) e parágrafos (“}”)
- C Substituir os caracteres da posição do cursor até ao fim da linha

:	Indica que os próximos comandos são comandos para o editor de linhas
:sh<Enter>	Regressar temporariamente à shell para executar comandos
^D	Retorno ao vi após edição de comandos na shell
:n<Enter>	Ir para a linha n do ficheiro
:x,zw fich<Enter>	Escrever para o ficheiro fich da linha x à linha z
.;Sd<Enter>	Apagar todas as linhas no <i>buffer</i> da corrente até à última
:r fich<Enter>	Inserir o conteúdo do ficheiro fich na linha corrente
:s/texto/palavras novas/<Enter>	Substituir a primeira ocorrência de “texto” com “palavras novas”
:s/texto/palavras novas/g<Enter>	Substituir todas as ocorrências de “texto” na linha corrente com “palavras novas”
:g/texto/s//palavras novas/g<Enter>	Substituir todas as ocorrências de “texto” por “palavras novas”
p	Posicionar o conteúdo do <i>buffer</i> temporário contendo a última eliminação ou comando de puxar para o texto, após o cursor
yy	Retirar uma linha de texto, colocando-a num <i>buffer</i> temporário
nyx	Retirar uma cópia de n objectos de texto x, colocando-os num <i>buffer</i> temporário
lyn	Colocar uma cópia do objecto de texto n no registo identificado pela letra l
lp	Colocar o conteúdo do registo l após o cursor

Comandos de abandono

- ZZ
- :wq<Enter> Escrever o ficheiro e abandonar o vi
 - :w fich<Enter> Escrever o *buffer* para um ficheiro chamado fich e abandonar o vi
 - :q<Enter> Escrever sobre o ficheiro chamado fich e abandonar o vi
 - :rw! fich<Enter> Abandonar o vi sem escrever o *buffer*
 - :q!<Enter> Abandonar o vi sem escrever o *buffer* para um ficheiro de sistema do Unix

Comandos especiais

- .
- Repetir o último comando
- J Juntar a linha abaixo com a linha corrente
- \x Colocar o carácter não imprimível x que não é possível colocar no modo de inserção de texto
- ^v Imprimir os caracteres que não são normalmente visíveis no modo de visualização de texto
- ! Limpar e redesenhar a janela actual
- Mudar entre maiúsculas e minúsculas

Opções especiais para o vi

- vi fich1 fich2 fich3 Coloca os ficheiros fich1, fich2 e fich3 no *buffer* do vi para serem editadas
- nv<Enter> Escreve o ficheiro actual e chama o ficheiro seguinte no *buffer*
- m<Enter> Recupera as alterações feitas a fich1
- vi -r fich1<Enter>

Posicionamento na janela

Posicionamento por carácter	
h	
<backspace>	← Desloca o cursor um carácter para a esquerda
l	
espaço	→ Desloca o cursor um carácter para a direita
fx	→ Move o cursor para a direita para o carácter x
Fx	← Move o cursor para a esquerda para o carácter x
;	↔ Segue para a próxima ocorrência da última pes- quisa
tx	→ Move o cursor para o carácter anterior à próxima ocorrência de x
Tx	← Move o cursor para o carácter seguinte à ocorrência anterior de x

Posicionamento por linha	
j	
÷	
<Enter>	↓ Desloca o cursor uma linha para baixo, mantendo-se, se possível, na mesma coluna
k	
-	↑ Desloca o cursor uma linha para cima, mantendo-se, se possível, na mesma coluna

Posicionamento por palavra	
w	→ Move o cursor para o primeiro caracter da palavra seguinte
W	→ Ignora toda a pontuação e move o cursor para a próxima palavra apenas delimitada por espaços em branco
b	← Move o cursor para o primeiro caracter da palavra anterior
B	← Move o cursor para a primeira palavra anterior delimitada apenas por espaços em branco
e	→ Move o cursor para o fim da palavra corrente
E	→ Delimita as palavras apenas por caracteres em branco. O cursor é colocado no último caracter antes do primeiro espaço em branco ou fim da linha
Posicionamento por frases	
(← Move o cursor para o início da frase corrente
)	→ Move o cursor para o início da frase seguinte
Posicionamento por parágrafos	
{	← Move o cursor para o início do parágrafo corrente
}	→ Move o cursor para o início da parágrafo seguinte
Posicionamento na janela	

H	↖ Move o cursor para o primeiro caracter da primeira linha no écran (<i>home</i>)
M	↔ Move o cursor para a linha a meio do écran
L	↘ Move o cursor para o primeiro caracter da última linha no écran

Posicionamento no ficheiro

Deslizamento	
^f	Desliza um écran para baixo
^d	Desliza meio écran para baixo
^b	Desliza um écran para cima
^u	Desliza meio écran para cima
^e	Desliza uma linha para baixo
^y	Desliza uma linha para cima
Posicionamento numa linha numerada	
G	Salta para a última linha do ficheiro
^g	Dá o número da linha e informação de estado
Pesquisa de padrões	
/padrão	Pesquisa para a frente a próxima ocorrência do padrão, posicionando o cursor no primeiro caracter
?padrão	Pesquisa para trás a próxima ocorrência do padrão, posicionando o cursor no primeiro caracter
n	Repete o último comando de pesquisa
N	Repete o último comando de pesquisa na direção oposta

Índice

Symbols

- <TAB>, 22
 - ΓA30C, 66
 - (, 55
 - (, 55
 - ←, 55
 - ^, 61
 - ^C, 17, 28, 32
 - ^D, 17, 32
 - ^Q, 17, 32
 - ^S, 17, 32
 - ^U, 17
 - ^backspace, 45
 - ^h, 55
 - ^↓, 48
 - ^←, 48
 - ^→, 48
 - ^↑, 48
 - E, 22, 31, 57
 - ~, 22
 - , 55
 - *, 7, 19, 31, 32, 70, 73
 - +, 36
 - „, 59
 - , 25, 34–36, 65
 - ., 18, 19, 21, 24, 25, 29, 31, 33, 47, 52, 60, 71
 - .., 21, 31
 - .Z, 67
 - .bash_profile, 19
 - .gz, 66
 - .joerc, 56
 - .login, 19
 - .mcwd, 72
 - .plan, 37
 - .profile, 19
 - .ssh/known_hosts, 64
 - .tar, 65
 - .tar.gz, 65
 - .tar.gz, 65
 - .zip, 67
 - /, 18, 20, 21, 31, 32, 71
 - /etc/group, 34
 - /etc/passwd, 37
 - :, 23, 37
 - ?, 20, 61, 70
 - \$, 24
 - &, 38
 - área, 3, 7, 15, 19, 23, 27, 39, 47, 69
 - écrans virtuais, 51
 - écrans virtuais
 - paginados, 47
 - écrans virtuais, 47, 51
 - paginados, 47, 48, 51
 - , 18
- ## A
- A:, 71, 72
 - account, *ver* login
 - administração, 7, 16, 18, 22, 27, 34, 38, 39
 - delegação, 18
 - permissões, 18
 - AfterStep, 51
 - Alt+Fn, 15, 17, 45
 - Anonymous, 69
 - AnotherLevel, 51
 - aplicação, 49
 - aplicações, 9, 43, 48–52, 63, 65, 66, 70
 - apropos, 29
 - arquivo, 61, 65–67, 73
 - ASCII, 33, 70
 - assinatura, 61
 - ataque, 63, 64

B

B:, 71
backspace, 55
backup, 65, 73
bash, 19, 23, 24
Blind Carbon Copy, *ver* bcc
Bourne Again Shell, *ver* bash
browser, 49, 59
browsers, 69
buffer, 57

C

C, 19, 28
C-shell, *ver* csh
Cópia de carbono, *ver* cc
cópia de ficheiros, 69
cópia de ficheiros, 69
caminho, *ver* pathname
case-sensitive, 7
cat, 31, 32, 35, 73
cavalos de Tróia, 3
cd, 21, 31
cd, 35, 70
CDE, 51
chat, 3
chave, 63, 64
chgrp, 34
chmod, 36, 37
cifra, 37, 63, 70, 71
cifrar, 71
colar, 50, 83
compilação, 38
compress, 67
compressão, 13, 65–67, 71
copiar, 50, 56
copy, *ver* copiar
core, 32
correio electrónico, 49
correio electrónico, 59, 60
correio electrónico, 37, 57, 59, 60
 assunto, *ver* subject
 attachments, 59
 bcc, 59
 cabeçalho, 59
 cc, 59
 corpo, 59
 subject, 59
 to, 59
correio electrónico

 bcc, 59
 cc, 59
correio-electrónico, 69
cortar, 50, 56, 83
cp, 33
crash do disco, 63
csh, 23
cshell, 19
Ctrl+Alt+Del, 17
Ctrl+Alt+Fn, 15
cut, *ver* cortar

D

dados
 lixo, 8
 persistência, 8, 15
Debian, 13
decifrar, 71
descompressão, 66, 67
directoria, 9, 18–25, 29–35, 51, 56,
 57, 65, 66, 69–72
 árvore, 20–22, 31, 49, 72
 corrente, 21–25, 31, 56, 66, 70–
 72
 HOME, 22, 31, 37, 47, 51, 56,
 71, 72
 raíz, 72
 raíz, 20, 21
 subdirectoria, 20, 31, 33, 65, 72
diskette, 71–73
diskettes, 71
DISPLAY, 49
download, 65
drive, 71, 72

E

e-mail, *ver* correio electrónico
edição de texto, 55, 57
editor de texto, 35, 56
emacs, 13, 31, 49, 57, 75
Esc, 55
exit, 17
export, 24
expressões regulares, 32
extensão, iii, 19, 20, 57, 65–67
 bas, 19
 c, 19
 h, 19
 p, 19
 txt, 19

F

ficheiro, 3, 8, 9, 15–25, 28–37, 47,
51, 52, 56, 57, 59, 64–67,
69–73, 79, 83–85
cópia, 33
correio electrónico, 59
Data/hora, 30
directorias, 18
eliminação, 32
executável, 19
fragmentação, 73
gestor, 57
grandes dimensões, 73
impressão, 33
nome, 18
renomeação, 33
salvaguarda e logout, 4
simples, 18
transporte, 73
fila de espera, 33
File Transfer Protocol, *ver* ftp
finger, 37
folder, 60
folders, 60
Free Software Foundation, 13
FSF, *ver* Free Software Foundation
ftp, 69, 70, 72
anónimo, 69
comandos, 70
ascii, 70
binary, 70
bye, 70
type ascii, 70
type binary, 70
user, 69
Modo de transferência, 70
pessoal, 69
ftp://, 69
fvwm, 51
fvwm2, 51
fvwm2-95, 51

G

gcc, 13
gdb, 13
ghostview, 49
gid, 37
GNU, 13, 66, 75
groups, 34

guest, 69
gzip, 13, 66, 71

H

hard links, 30
HOME, 22, 23
host, 69
http://, 69

I

Inbox, 60
Internet, 49, 59, 69
endereços, 69
interpretador de comandos, *ver* shell

J

janela, 7–9, 41, 43, 45, 47–49, 51,
79, 84, 85
activa, 48, 50, 52
gestores, 51, 52
janela activa, 51
joe, 56, 79
ajuda, 56
configuração, 56
Joe's Own Editor, *ver* joe
joerc, 56
Joseph H. Allen, 56

K

KDE, 51
kill, 38

L

less, 17, 28, 31, 32
LILO, *ver* Linux Loader
link simbólico, 30
Linux, iii, 5, 9, 13, 15–18, 28, 38, 45,
48, 49, 56, 60, 71
Linux Loader, 5
Linux Loader, 5, 8, 9, 17
livro de endereços, 60
login, iv, 4, 7, 9, 15, 16, 23, 37, 45,
47, 63
erros, 16
incorrecto, 16, 45
remoto, 63
logout, iv, 4, 7, 17
lpq, 34
lpr, 25, 33

lprm, 34
ls, 18–21, 25, 29–31, 35
ls, 70

M

man, 22, 28, 29
manual, 23, 28, 29, 66
 secção Description, 28
 secção Name, 28, 29
 secção See Also, 28
 secção Synopsis, 28
 secções, 28
mcd, 72
mcopy, 72
mdel, 72
mdeltree, 72
mdir, 72
menu, 7, 45, 48, 49, 51, 57, 61
 Start, 7, 48
meta-caracteres, 19, 20, 32, 33, 70,
 71
 *, 19, 32, 70, 73
 ?, 20, 70
mformat, 72
mkdir, 31
mmd, 72
mmove, 72
more, 17, 28, 31, 32, 35
mpage, 33
mrd, 72
mren, 72
MS-DOS, 19, 71, 72
mtools, 71, 72
mtype, 72
mv, 33

N

Netscape Navigator, 49
Network File System, 15
Network Neighborhood, 9
newsgroup, 3, 59–61
 post, 60
NFS, *ver* Network File System

O

octal, 35, 36
olvwm, 51
olwm, 51

P

Page Down, 56
Page Up, 56
palavra-passe, *ver* password
Pascal, 19
passwd, 27
password, 3, 4, 7, 16, 18, 27, 28, 37,
 45, 61, 63, 69–71
 incorrecta, 7
 regras, 4
paste, *ver* colar
PATH, 23–25
path, *ver* pathname
pathname, 21, 22, 31, 33, 71
 <TAB>, 22
pdf, 49
permissão, 18, 19, 22, 30, 31, 35–37,
 69
 dono, 30, 34–37
 escrita, 30, 34–37
 execução, 18, 30, 31, 34–37
 execute, 30
 group, *ver* grupo
 grupo, 30, 34–36
 leitura, 30, 34–37
 other, *ver* outros
 outros, 30, 34–36
 owner, *ver* dono, *ver* dono
 read, 30
 representação numérica, 35
 write, 30
pesquisa, 32, 56
pine, 60, 61
pipe, 65, 66
postscript, 33, 49
printf, 28
processo, 9, 18, 24, 28, 38
 background, 38, 49
 pid, 38
 terminar, 38
prompt, 17, 21–23, 38
protecção, 34, 36
 grupo, 36
 outros, 36
 representação numérica, 36, 37
 todos, 36
 user, 36
ps, 38
pwd, 31

Q

quota, 39
 quota, 39

R

rato, 43, 48, 57, 69
 botão esquerdo, 50
 botão 3, 50
 botão direito, 50
 botão esquerdo, 48
 ecran virtual, 47, 48
 janela activa, 51
 menu, 48
 selecção de texto, 50
 read-only, 35
 Red Hat, 13
 redundância, 66
 regras, 3, 60
 newsgroups, 60
 reinicialização, 4, 8, 9, 16–18, 45
 rm, 20, 32
 rmdir, 31, 33
 root, *ver* administração

S

scp, 69–72
 secure copy, *ver* scp
 secure shell, *ver* ssh
 segurança
 quebra, 25
 servidor, 9, 16, 38, 43, 59, 61, 63,
 64, 69–72
 sessão, 4, 8, 9, 15–17, 21, 23, 38, 45,
 47, 49, 69–72
 modo de texto, 15, 17, 43, 45,
 49, 56, 60
 remota, 49
 set, 24
 shell, iv, 19, 22–24, 37, 38, 43, 45,
 49, 60, 63, 71, 79
 variável, 23
 Shut down, 7
 shutdown, 9
 sistema de ficheiros distribuído, 39
 sistema de ficheiros distribuído, 15
 Sistema de ficheiros distribuído, 72
 Slackware, 13
 software
 instalação, 3

split, 73
 ssh, 63
 problemas, 63
 ssh, 39, 49, 63
 ssh, 43, 63, 64, 70, 71
 stdin, 65
 stdout, 65, 66
 substituição, 56

T

Tape Archive Utility, *ver* tar
 tar
 adicionar, 65
 compressão, 66
 criação, 65
 criar, 65
 descompressão, 66
 e compressão, 65
 extracção, 65, 66
 extrair, 65
 listar, 65
 testar, 65
 tar, 65, 66
 tar, 65, 66, 71
 TC-Shell, *ver* tcsh
 tcsh, 23
 telnet, 63
 TheNextLevel, 51
 twm, 51

U

uid, 37
 uncompress, 67
 unidades de banda, 65
 UNIX, 19
 Unix, iii, iv, 11, 13, 15, 17–20, 22–
 25, 27–29, 31, 32, 34, 36,
 38, 39, 43, 52, 55, 65, 67,
 69–72, 84
 unset, 24
 unzip, 67
 username, 7, 63, 69
 users, 37
 utilizadores
 adicionar, 18
 eliminar, 18

V

vírus, 3
 vi, 31, 55, 56, 81–84

Visual Editor, *ver* vi

W

w, 37

whatis, 29

who

 who am i, 37

who, 37

wildcards, *ver* meta-caracteres

WindowMaker, 51

Windows, 48, 66, 67

Windows 95, 7, 48, 51

Windows 98, 7

Windows NT, 5, 7–9, 15

winzip, 66, 67

X

X Windows, 15, 49, 50, 60

X Windows, iii, 15, 41, 43, 45, 48–
51, 57, 63

xemacs, 57

xhost, 49

Z

zip, 67

Bibliografia

- [Art90] Lowell Jay Arthur. *Unix Shell Programming*. Number ISBN 0-471-51820-4. John Wiley & Sons, Inc., New York, USA, 1990.
- [BBM85] Peter Birns, Patrick Brown, and John C. C. Muster. *Unix for people: A Modular Guide to the Unix Operating System: Visual Editing, Document preparation & Other Resources*. Number ISBN 0-13-937459-0. Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632, USA, 1985.
- [Bou83] S. R. Bourne. *The Unix System*. Number ISBN 0-201-13791-7 in International Computer Science Series. Addison-Wesley Publishing Company, 1983.
- [PDMM90] Carlos Miguel Picoto, Henrique João L. Domingos, José A. Legatheaux Martins, and José Carlos Monteiro. Introdução à utilização do Sistema UNIX v. 1.0. Technical Report DICC-FCUL-90-01, Departamento de Informática - Faculdade de Ciências - Universidade de Lisboa, 1990.
- [TS87] Grace Todino and John Strong. *Learning the Unix-Operating System*. Number ISBN: 0-937175-16-1. O'Reilly & Associates, Inc., 1987.