

Curso de Linux e Redes

Núcleo Universitário do IEEE

26 de Setembro de 2002

Conteúdo

1	Introdução	3
2	Ambiente Informático da FEUP	4
2.1	Windows	4
2.1.1	Introdução	4
2.1.2	Sistema Operativo	4
2.1.3	Serviços do CICA	13
2.2	Linux	20
2.2.1	Introdução	20
2.2.2	Ficheiros	21
2.2.3	X Windows	26
2.2.4	Consola	28
2.3	TCPGATE	40
2.4	Página Pessoal	41
2.5	GAUTI	43

Agradecimentos

Para a realização deste curso foi necessária a participação de inúmeras pessoas às quais gostaríamos de agradecer:

- Professor Carlos Costa
- Professor Fernando Nunes Ferreira
- Professor Silva Matos
- Eng. Susana Gaio
- todos os formadores
- todos os monitores

Queríamos agradecer ainda às seguintes instituições pelos seu apoio e disponibilização de material e espaço:

- CICA
- Departamento de Engenharia Electrotécnica e Computadores
- Gaudi
- Serviços Académicos e de Recursos Humanos - Unidade de Pré-Graduação

Capítulo 1

Introdução

Mais uma vez se organiza o Curso de Linux e Redes, cabendo este ano ao Núcleo Universitário do IEEE promovê-lo.

Ao longo dos últimos anos o curso tem vindo a adaptar-se às necessidades de quem entra pela primeira vez para a FEUP, bem como aos recursos que estão disponíveis para a população académica. Assim, este ano optou-se por incluir no curso uma pequena parte dedicada ao ambiente informático da FEUP. Nesta sebenta serão discutidos o sistema Windows disponibilizado pelo CICA, o serviço de TCPGATE e será feita uma pequena apresentação do Gabinete de Apoio à Utilização das Tecnologias de Informação (GAUTI).

Obviamente a parte central do curso será a introdução ao sistema operativo Linux o que implica também que se expliquem alguns conceitos breves sobre Redes.

O objectivo desta sebenta é não só ajudar os formadores e alunos durante as sessões do curso, mas também de avançar um pouco mais, apoiando o leitor nas suas primeiras incursões no ambiente informático da FEUP.

Capítulo 2

Ambiente Informático da FEUP

2.1 Windows

2.1.1 Introdução

Tal como foi dito anteriormente o Windows é um dos sistemas operativos (S.O.) colocado à disposição pelo CICA nas estações de trabalho da faculdade.

Apesar da versão 2000 ser muito parecida com as versões utilizadas normalmente por utilizadores domésticos (95/98/ME) a nível de interface, contém funcionalidades que não estão presentes neste, uma vez que é um S.O. mais vocacionado para sistemas em rede, contendo funções específicas para a mesma, e para vários utilizadores, sendo, por exemplo, necessário introduzir um nome de utilizador e uma *password* para iniciar uma sessão de trabalho.

Como tal apresentaremos a seguir algumas noções (possivelmente novas para a maior parte dos alunos), funcionalidades do S.O. não presentes nos sistemas caseiros, e serviços fornecidos pelo CICA neste ambiente.

2.1.2 Sistema Operativo

Profile

O *profile* é o ambiente de trabalho do utilizador, ou seja, os ficheiros característicos de cada utilizador para executar uma sessão de trabalho, com as suas preferências não só a nível de interface gráfica como a nível de programas, por exemplo, os *sites* favoritos da Internet, as *cookies* guardadas pelo navegador de web, livro de endereços, etc.

Sendo o *profile* característico de cada utilizador e contendo informações necessárias para executar uma sessão de trabalho, é necessário que mudando de computador o *profile* se mantenha igual, ou seja, é preciso que a informação armazenada num computador seja transportada para o novo computador no qual o utilizador vai executar outra sessão, de uma forma transparente para o utilizador. Este processo é

executado pelo S.O. que no final de cada sessão copia o profile local com as preferências do utilizador para um servidor central, sendo este depois copiado a partir do servidor para o novo computador no qual o utilizador vai executar uma nova sessão de trabalho. Este tipo de profile denomina-se por *Roaming profile*.

O profile está localizado em `C:\Documents and Settings\<login do utilizador>\`. Dentro desta directoria existe uma hierarquia de directórios que contém os ficheiros do utilizador. De seguida apresentam-se as directorias que consideramos mais importantes e uma breve descrição do seu conteúdo:

Cookies\ Esta directoria guarda as *cookies* que o browser vai recebendo dos vários *sites* de Internet visitados. Estes pequenos ficheiros de texto permitem que as diversas paginas visitadas guardem informação específica sobre o utilizador permitindo que certas funcionalidades se mantenham da próxima vez que for visitada (cor de fundo, preferências, ultimas buscas, etc).

Desktop\ Esta directoria contém os ficheiros que aparecem no ambiente de trabalho. Aqui podem ser colocados atalhos para os programas mais utilizados ou ficheiros temporários.

Favorites\ Esta directoria contém os sítios de Internet favoritos que foram adicionados ao Internet Explorer.

My Documents\ Esta directoria contém os documentos pessoais do utilizador.

Start Menu\ Aqui podem ser acrescentados atalhos para as aplicações. Desta maneira o utilizador pode personalizar o menu de *Start* colocando atalhos para as aplicações que usa mais.

Application Data\ Este directório é utilizado por algumas aplicações para guardar informação específica sobre o utilizador, tais como ficheiros resultantes da navegação na Internet (Mozilla e Netscape), os computadores acedidos através do cliente de SSH, etc).

Local Settings\ Este directório também é utilizado por algumas aplicações para guardar informação específica sobre o utilizador (no subdirectório escondido "Application Data"). Contem ainda os ficheiros temporários utilizados pelo S.O. ao longo da sessão de trabalho do utilizador e o histórico dos sites de Internet visitados pelo utilizador.

Todos os ficheiros contidos nestes directório são específicos do utilizador e como tal são transferidos para o servidor central quando a sessão de trabalho é encerrada.

Passwords

Para iniciar uma sessão nas estações de trabalho da faculdade é necessário um nome de utilizador e a respectiva password. Desta maneira protegem-se os dados de cada utilizador e os serviços a que o mesmo tem acesso.

A escolha de uma *password* segura é **muito importante** uma vez que o utilizador é responsável pela segurança mesmo que essas acções tenham sido executadas por outra pessoa. É da responsabilidade do utilizador escolher uma *password* segura de maneira a evitar acessos não autorizados à sua conta. Assim a password deve seguir pelo menos duas das seguintes normas de maneira a tornar mais difícil a sua descoberta:

- Ter pelo menos 8 caracteres.
- Conter números
- Conter maiúsculas
- Conter caracteres especiais (por exemplo: !*"#\$%)
- etc

Um outro factor importante na segurança das passwords é a frequência com que é mudada. Quanto maior for a frequência mais segura a conta está. Como tal deve mudar a password pelo menos 1 vez por mês.

Em Windows 2000 a password pode ser mudada executando os seguintes passos:

1. pressionar + + .
2. Pressionar **Change Password**.
3. Preencher os campos **Old Password**, **New Password** e **Confirm New Password**.
4. Pressionar **OK**.

Recuperar de Encravamentos

Já se sabe que sendo o software feito por humanos está sujeito à introdução de erros, os famosos *bugs*, que acabam por resultar em encravamentos do S.O.

O Windows 2000 já é um sistema bastante maduro sendo os encravamentos a nível do S.O. extremamente raros. A grande maioria dos encravamentos ocorrem a nível dos programas que são utilizados. Devido ao sistema de execução em "bolha" o encravamento de uma aplicação não afecta a estabilidade de outras aplicações ou do S.O., sendo possível continuar a sessão de trabalho após o encerramento da aplicação encravada.

Para encerrar uma aplicação encravada executar os seguintes passos:

1. pressionar **Ctrl** + **Alt** + **Del**.
2. pressionar **Task Manager**.
3. seleccionar o separador de aplicações.

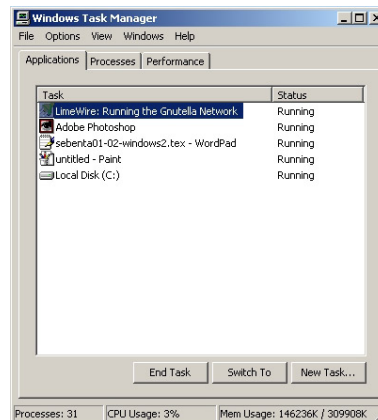


Figura 2.1: Janela do *Task Manager*.

4. seleccionar a aplicação encravada.
5. pressionar o botão **End Task**.

Assim quando o utilizador pensa estar perante um encravamento devido à falta de resposta do sistema **NAO FAÇA RESET!!!** O Windows enquanto em funcionamento contém inúmeros ficheiros de sistema abertos que com o reset não são correctamente fechados, ficando corrompidos e resultando na destruição do sistema.

Gostaríamos de alertar para o facto de num sistema em rede existirem várias comunicações entre a estação de trabalho e mais do que um servidor. Muitas vezes, devido a congestionamentos da rede ou a um período de inactividade do servidor, o S.O. deixa de responder aos comandos do utilizador enquanto espera a resposta do mesmo, parecendo um encravamento quando na verdade não o é. Passado alguns segundos o computador começa a responder quando recebe a resposta ou quando se apercebe que o servidor não está operacional. Um exemplo disto é, por exemplo, quando se tenta aceder a um computador da rede que esta desligado ou está em Linux. Normalmente o Windows fica cerca de um minuto à espera de resposta até exibir a mensagem de erro. Durante este período o sistema perde alguma capacidade de resposta a nível do explorador podendo parecer que está encravado.

Montar sistemas de ficheiros

Sendo o Windows 2000 um sistema operativo vocacionado para redes contém funcionalidades para facilitar o acesso aos vários computadores da mesma. Como tal é

possível montar uma *drive* ou directório remoto, ou seja, fazer com que um directório ou computador da rede apareça no tradicional *My Computer* como uma *drive*, tal como o C:\ ou A:\ , facilitando o acesso ao computador e aos ficheiros remotos.

O *profile* de cada utilizador na FEUP já vem configurado automaticamente para montar a drive Z:\ que permite aceder aos ficheiros da conta presente no *cluster*¹ UNIX do CICA (ver mais à frente em serviços).

Para montar uma *drive* executar os seguintes passos:

1. Abrir o **My Computer**.
2. Aceder ao menu **Tools -> Map Network Drive**.

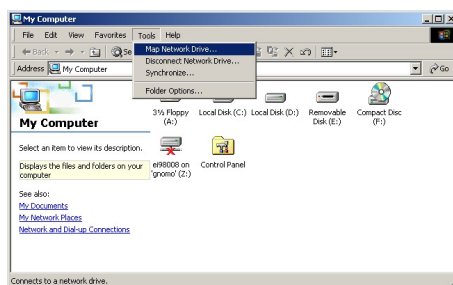


Figura 2.2: Aceder a menu **Tools -> Map Network Drive**.

3. Escolher a letra para a drive (não pode se uma letra já mapeada).
4. Escolher o computador ou directório remotos escrevendo o caminho em **Folder** ou seleccionando na caixa de diálogo depois de pressionar **Browse**.

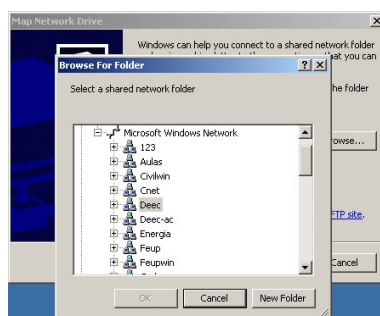


Figura 2.3: Escolher directório ou computador através da opção **Browse**

5. Decidir se se pretende que a drive seja montada na próxima vez que iniciar uma sessão de trabalho preenchendo (ou não) o campo **Reconnect at logon**.
6. pressionar o botão **Finish**.

¹Conjunto de máquinas ligadas entre si destinadas a fornecer serviços a um grande número de utilizadores e que funcionam como uma só máquina.

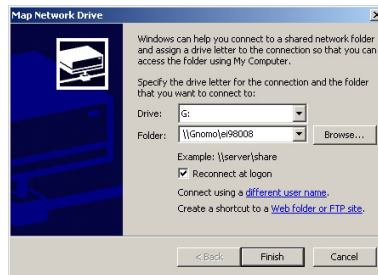


Figura 2.4: Janela preenchida.

Aceder a computadores da Rede

Na faculdade o utilizador tem ao seu dispor um grande número de computadores para utilizar. No entanto existe também um grande número de alunos e, como tal, uma constante rotação na ocupação dos mesmos. Mas, durante uma sessão de trabalho, são muitos os ficheiros manipulados e criados, muitas vezes descarregados da internet, que apenas ficam localmente no computador a ser utilizado. Na próxima vez que o utilizador pretender aceder aos ficheiros é necessário que o **mesmo** computador esteja vago o que muitas vezes não acontece.

Para contornar este problema os computadores das salas de informática do CICA estão partilhados, ou seja, estão acessíveis a partir de outros computadores, podendo o utilizador aceder aos ficheiros que se encontram num computador remoto (atenção que apenas a *drive D:* de cada computador está acessível desta forma, ver mais à frente em serviços).

Esta funcionalidade também é prática para partilhar recursos apenas disponíveis em alguns computadores, como por exemplo, a Zip Drive, permitindo copiar ficheiros locais de um computador sem Zip para um computador remoto com Zip na mesma sala de informática, na sala ao lado, ou do outro lado da Faculdade.

Para aceder aos computadores partilhados em rede, executar os seguintes passos:

1. Abrir **My Network Places**

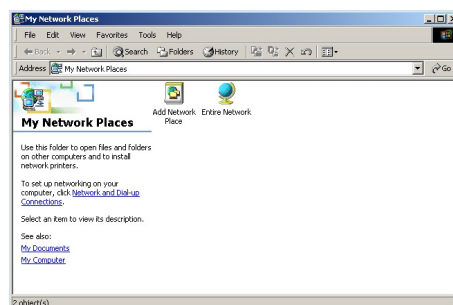


Figura 2.5: Janela de My Network Places.

2. Seleccionar **Entire Network**.

3. Selecionar **entire contents** no lado esquerdo da janela.

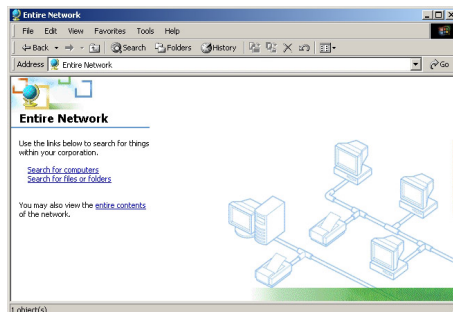


Figura 2.6: Selecionar entire contents.

4. Selecionar **Microsoft Windows Network**.

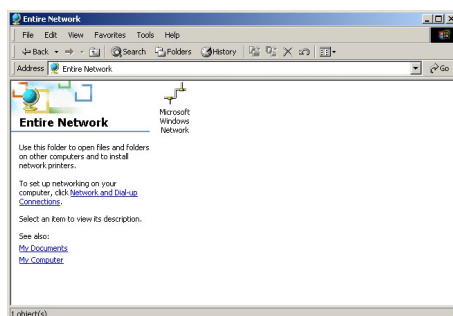


Figura 2.7: Selecionar Microsoft Windows Network.

5. Selecionar a sub-rede onde se encontra o computador (a grande maioria dos computadores das salas destinadas a alunos encontra-se na sub-rede **Aulas**, os computadores do cluster UNIX - tom, alf, gnomo, yoda, etc - encontram-se na sub-rede **FEUPsig** e os computadores das salas do DEEC encontram-se na sub-rede **Deec**).

6. Selecionar o computador desejado.

ou alternativamente, e sabendo o nome ou IP do computador a que se pretende aceder:

1. Ir ao menu de **Start** e escolher **Run**.

2. Colocar o nome ou IP da seguinte forma: \\nome-do-computador ou \\IP.

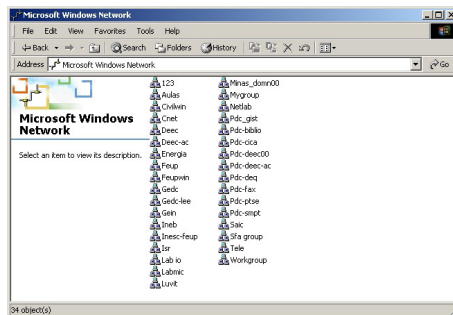


Figura 2.8: Seleccíonar Sub-rede.

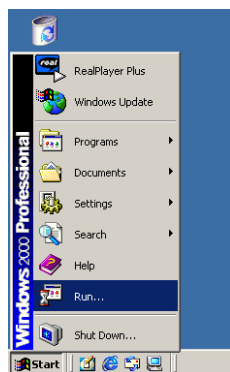


Figura 2.9: Menu Start -> Run.

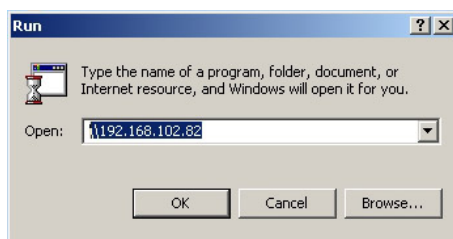


Figura 2.10: Aceder a computador usando o IP.

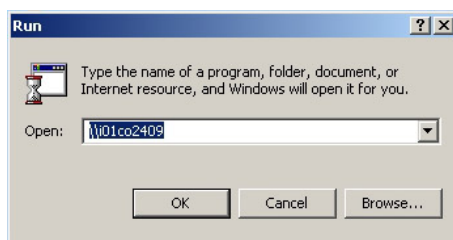


Figura 2.11: Aceder a computador usando o nome do computador.

Permissões e instalações

Para proteger os sistemas de erros dos utilizadores (por exemplo apagar ficheiros vitais ao sistema "acidentalmente") e da instalação de programas que tornem o sistema instável (por exemplo dois programas que executem exactamente as mesmas tarefas, tal como o anti-vírus) o Windows fornece um sistema de controlo de permissões, ou seja, permite definir, utilizador a utilizador, qual o seu campo de acção, o que pode e não pode executar, fazer ou ver. Assim no sistema da faculdade, e pelas razões apontadas acima, os alunos não têm permissões para instalar programas adicionais nem escrever ficheiros para o disco C:\ . Se o aluno necessitar escrever ficheiros localmente deve fazê-lo para o D:\ (disco temporário fornecido pelo CICA, ver mais à frente na secção "Disco temporário"), e se necessitar de instalar um programa deve falar com o CICA para saber se é possível a sua instalação ou não.

Net Send

Sendo o Windows 2000 um S.O. vocacionado para sistemas multi-utilizador, no qual existem vários utilizadores a partilhar uma rede local, é fornecido pelo S.O. um mecanismo simples de comunicação entre utilizadores numa mesma rede: o `net send`.

Através deste comando é possível enviar uma mensagem a outro utilizador com uma sessão de trabalho aberta e que pertence à mesma rede. Caso o utilizador não esteja ligado é emitida uma mensagem informativa do facto podendo assim este comando ser utilizado para verificar se um dado utilizador está ligado à rede.

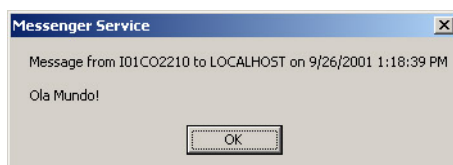
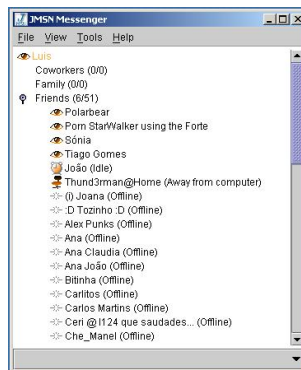


Figura 2.12: Mensagem através do net send.

No entanto este serviço foi desactivado dos computadores. Como alternativa para comunicação com outros utilizadores podemos sugerir os serviços de *Instant Messaging*. Do vasto leque de opções salientamos o Microsoft Messenger que se encontra instalado em todos os computadores e o JMSNmessenger, um cliente multiplataforma escrito em Java



Microsoft Messenger.



JMSNmessenger.

2.1.3 Serviços do CICA

Proquota

Podendo o *profile* conter ficheiros do utilizador além das preferências, e tendo este que ser copiado no início e no fim de cada sessão através da rede, o CICA adoptou um sistema de quotas para o *profile*. Isto permite diminuir o tempo da transferência do mesmo no fim e início de cada sessão, evitar uma carga excessiva da mesma transferência sobre a rede, e diminuir o espaço ocupado nos servidores centrais (uma vez que a FEUP contém cerca de 5000 alunos).

Assim o *profile* está limitado a 2 MB através de um sistema denominado Proquota, que avisa quando o profile está cheio e impede o encerramento da sessão no caso de o limite ter sido excedido. Neste último caso é necessário proceder à remoção de alguns ficheiros do profile nos directórios referidos na secção "Profile". Para ser mais fácil saber quais os ficheiros que estão a ocupar mais no *profile*, e qual o tamanho do profile nesse momento, o Proquota exhibe uma pequena janela com os ficheiros do profile por ordem decrescente de tamanho (ou seja o ficheiro maior aparece no topo pois este é o principal candidato à remoção) e o tamanho do *profile*.



Figura 2.13: Icon do proquota no *System Tray* quando o espaço suficiente.



Figura 2.14: Icon do proquota no *System Tray* quando o espaço é reduzido.



Figura 2.15: Icon do proquota no *System Tray* quando o espaço foi excedido.

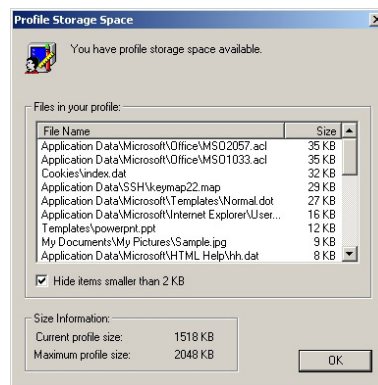


Figura 2.16: Janela do proquota com informações sobre o estado do profile.

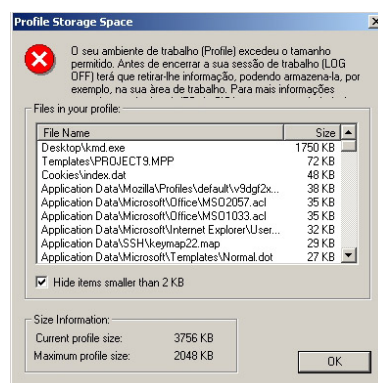


Figura 2.17: Janela do proquota quando o espaço foi excedido.

Discos temporários

O CICA equipou as estações de trabalho com um disco adicional ao disco principal C:\ identificado pela drive D:\. Este disco está à total disposição dos utilizadores para a colocação de ficheiros necessários à execução de uma sessão de trabalho. No entanto, sendo este disco limitado a 500 MB e acessível a qualquer utilizador que inicie uma sessão nesse computador, é necessário a **formatação diária** (efectuada entre as 7:00 e as 8:00) dos discos, de modo a evitar a falta de espaço devido ao enchimento do mesmo. Assim todos os ficheiros colocados neste disco devem ser apenas temporários, tendo que ser copiados para outro suporte (Zip ou conta UNIX) antes do fecho da sessão.

Conta UNIX

Para a colocação de ficheiros importantes e pessoais (documentos privados, trabalhos, relatórios), o CICA põe à disposição de cada utilizador uma conta no *cluster*² UNIX alojado no CICA. Esta conta tem o espaço de 10 MB e é automaticamente montada no início de cada sessão de trabalho sendo identificada pela letra Z:\.

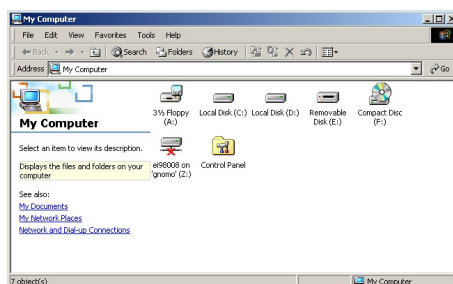


Figura 2.18: *Drive Z:\ no Meu computador.*

Caso a *drive Z:* se encontre indisponível devido a uma falha no servidor, é possível aceder à conta através de outros computadores pertencentes ao cluster:

- tom
- alf
- jerry
- crazy
- yoda
- gnomo

²Conjunto de máquinas ligadas entre si destinadas a fornecer serviços a um grande número de utilizadores e que funcionam como uma só máquina.

Para aceder à conta através de outro computador do cluster pode montar uma nova *drive* tal como foi descrito na secção "Montar Sistemas de ficheiros" ou aceder directamente através do comando Start -> Run:

1. Ir ao menu de **Start** e escolher **Run**.

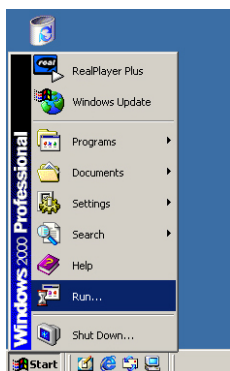


Figura 2.19: Menu Start -> Run.

2. Escrever `\\nome-do-computador\login`.

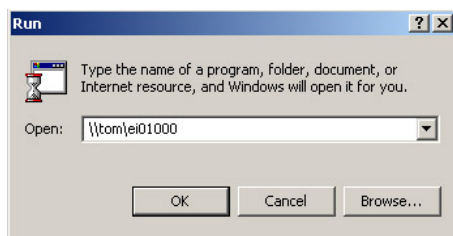


Figura 2.20: Aceder a conta Unix através de um dos computadores do *cluster*.

O cluster também fornece serviços de FTP, podendo a transferência de ficheiros para a conta UNIX ser feita através deste serviço caso não seja possível aceder à conta através dos métodos referidos anteriormente.

Imprimir

O CICA fornece aos alunos serviços de impressão permitindo que os alunos imprimam formulários, acetatos de aulas ou até documentos pessoais.

O serviço de impressão é efectuado baseado num sistema de quotas, no qual cada aluno compra as impressões ao CICA sendo-lhe atribuído uma quota igual ao montante comprado. A cada impressão é então descontado na quota o custo da impressão baseado no número de páginas impressas. Caso a quota seja nula o aluno pode recarregá-la em qualquer momento dirigindo-se ao HelpDesk do CICA.

Anualmente o CICA fornece a cada aluno 250 páginas de quota gratuitamente, podendo esta ser recarregada no helpdesk do CICA

O sistema de quotas é totalmente automático e transparente para o utilizador bastando executar o comando da impressão da aplicação que está a correr.

Pode-se visualizar o estado de todas as impressoras `http://pserver.fe.up.pt` ou em alternativa a fila de impressão da impressora k será\foi usada para imprimir em `http://pserver.fe.up.pt:90`.

Zip

Muitas vezes torna-se necessário transportar ficheiros da faculdade para casa ou outro local, tendo que se proceder à cópia desses mesmos ficheiros para um suporte que permita o seu transporte. O suporte tradicional é a disquete de 3 1/2, mas o CICA fornece nas salas de informática a opção de utilização de disquetes Zip 250MB/100MB, que permitem o transporte de um maior volume de dados. Em cada sala existem normalmente 2 *drives* Zip, sendo necessário utilizar a rede para transferir dados, caso a estação de trabalho não contenha uma unidade. O procedimento que a seguir se descreve funciona apenas se o computador no qual se encontra a *drive* estiver em Windows:

1. Abrir **My Network Places**.

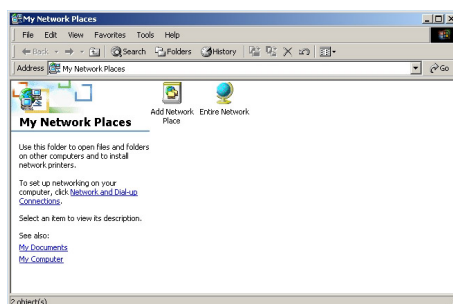


Figura 2.21: Janela de My Network Places.

2. Seleccionar **Entire Network**.
3. Seleccionar **entire contents** no lado esquerdo da janela.
4. Seleccionar **Microsoft Windows Network**.
5. Seleccionar a sub-rede onde se encontra o computador (a grande maioria dos computadores das salas destinadas a alunos encontra-se na sub-rede **Aulas**, os computadores do cluster UNIX - tom, alf, gnomo, yoda, etc - encontram-se na sub-rede **FEUPsig** e os computadores das salas do DEEC encontram-se na sub-rede **Deec**).

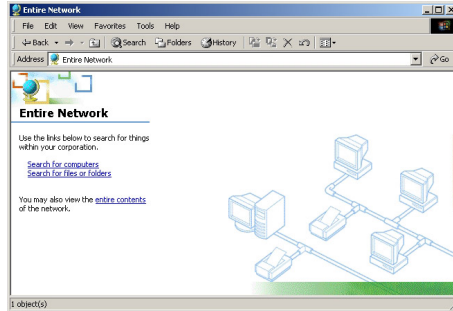


Figura 2.22: Seleccionar entire contents.

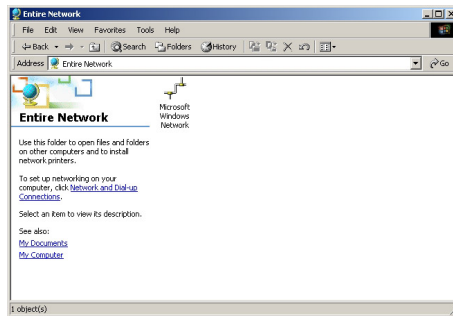


Figura 2.23: Seleccionar Microsoft Windows Network.

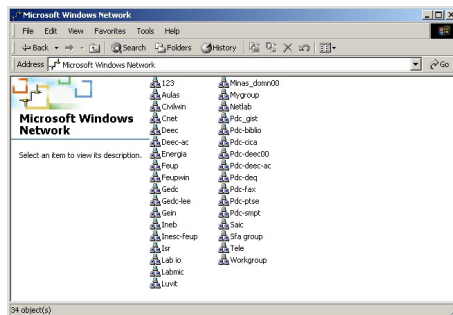


Figura 2.24: Seleccionar Sub-rede.

6. Seleccionar o computador com Zip.
7. Abrir o directório **E** ou **Zip** dependendo do nome do directório partilhado.
8. Copiar

ou alternativamente:

1. Ir ao menu de **Start** e escolher **Run**.

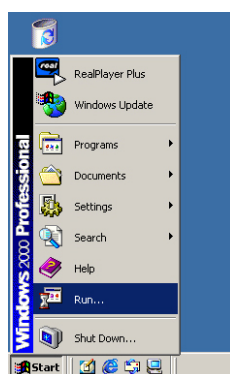


Figura 2.25: Menu Start -> Run.

2. Escrever \\nome-do-computador-com-zip.

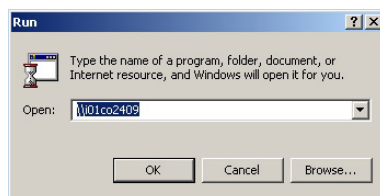


Figura 2.26: Aceder a computador computador com Zip.

3. Abrir o directório **E** ou **Zip** dependendo do nome do directório partilhado.
4. Copiar

2.2 Linux

2.2.1 Introdução

O que é o Linux

O Linux é um sistema operativo criado por Linus Torvalds em 1991 e actualmente desenvolvido por milhares de pessoas em todo o mundo. Segue a norma Unix, mas é orientado para máquinas de pequeno porte, tipicamente microcomputadores pessoais, embora seja cada vez mais frequente encontrar servidores de médio porte a correr Linux.

Em que é que se usa Linux

O Linux é actualmente utilizado em larga escala em ambiente universitário, empresarial e governamental. Para as universidades usar Linux é importante, não só pelo facto de não terem de pagar licença para o possuírem (é um sistema operativo completamente gratuito), mas também pela oportunidade de aprendizagem inigualável que proporciona (é um sistema operativo *Open Source*, ou seja, qualquer pessoa pode ver por si mesma como é que o sistema está construído e “aprender mexendo”). O Linux também é utilizado em empresas que necessitam de elevada performance, fiabilidade e segurança. Tipicamente as empresas de telecomunicações e tecnologias de ponta. O Linux, devido à indiscutível segurança que proporciona, é cada vez mais utilizado pelos governos ou agências governamentais que necessitam de manter informação simultaneamente disponível e segura. Nos últimos anos o Linux tem-se tornado um sistema operativo popular, sendo já possível encontrá-lo em casa de pessoas que tenham alguma familiaridade com a informática.

Distribuições

Uma distribuição é o núcleo do Linux (*kernel*) mais um vasto conjunto de aplicações que servem de suporte ao S.O. e ao utilizador. Existem várias boas distribuições de Linux, tendo cada uma as suas características particulares, vantagens e desvantagens. Algumas apostam em incluir inúmeras aplicações, outras, pelo contrário, optam pelo minimalismo; algumas apostam na automatização de procedimentos (pensando nos utilizadores pouco experientes), outras são mais orientadas para veteranos; mas, na sua essência, todas se baseiam no mesmo núcleo o (*kernel*!). Cabe pois a cada utilizador decidir qual a distribuição que mais lhe convém para o uso que vai fazer da sua máquina.

Algumas das distribuições mais conhecidas (listadas sem nenhuma ordem particular) são: SuSE, Slackware, Debian, RedHat, Caldera, Mandrake...

Nota: Se estiver interessado em fazer o download de uma distribuição poder-se-á dirigir a <http://www.linux.com> ou <http://linux-iso.org>.

Vantagens e Desvantagens

Quando se fala de vantagens e desvantagens do Linux é inevitável estabelecer comparações com outros sistemas operativos, nomeadamente com os sistemas operativos da Microsoft. A grande vantagem do Linux em relação a esses sistemas é ser *Open Source*, quase todas as outras derivam daí. Ser *Open Source* significa que todo o código é livre e disponibilizado em conjunto com a versão compilada. Não há segredos num software *open source*, todo o código é visível, e por isso é fácil encontrar erros e falhas. Por outro lado, qualquer pessoa com alguns conhecimentos de programação pode acrescentar o seu pedaço de código, contribuindo para que evolua diariamente. Desta forma, é cada vez mais raro encontrar falhas de segurança ou de outro tipo em sistemas Linux, e quando são encontradas são rapidamente corrigidas. O Linux torna-se assim mais rápido, mais fiável e mais seguro que o tradicional Windows. Uma desvantagem inerente é o facto de o Linux ter sido desenvolvido essencialmente entre programadores ou pessoas com conhecimentos avançados de informática, o que significa que não tem tantas preocupações com o utilizador inexperiente como os sistemas da Microsoft. Esta questão está hoje em dia praticamente ultrapassada, uma vez que há algumas distribuições de Linux que se dedicaram a produzir um interface mais fácil e intuitivo, muitas vezes semelhante ao do Windows. Alguém uma vez disse: "Com o Linux temos a certeza que somos nós a controlar o sistema operativo, e não o sistema operativo a controlar-nos a nós". Assim é.

“E quanto ao meu sistema operativo antigo?”

Existem duas formas de preservar o seu sistema operativo antigo, e até mesmo usá-lo se for necessário. A forma mais simples e eficaz é instalar um pequeno programa (chamado `lilo` - o *linux loader*), que questiona o utilizador sobre o sistema operativo a usar quando o sistema arranca. Isto permite-lhe ter o seu sistema antigo intacto numa partição diferente, sem ter o mínimo de problemas (o outro sistema operativo nem saberá que o Linux está instalado!).

A outra forma é usando um emulador ou simulador. Podemos mencionar o `wine`, um simulador que permite correr algumas aplicações do sistema Windows, `wineX` similar ao anterior mas com algum suporte para DirectX (jogos!) ou até mesmo o `VMware` (comercial) que virtualiza completamente uma máquina, permitindo assim instalar e correr qualquer sistema operativo sobre ela.

2.2.2 Ficheiros

Permissões e Tipos de Utilizadores

Em Unix/Linux cada ficheiro tem associado um conjunto de atributos de segurança, que determinam o tipo de acesso que cada utilizador tem ao ficheiro. A este conjunto de atributos dá-se usualmente o nome `permissões`.

Cada ficheiro possui atributos de leitura (para ler o conteúdo do ficheiro), escrita (para modificar ou apagar o ficheiro) e execução (para executar um programa ou script). Na realidade, cada ficheiro tem três conjuntos destes três atributos, um por cada tipo de utilizador.

Em Unix/Linux existem três tipos de utilizadores distintos no que diz respeito a manipulação de ficheiros. O “owner” é o dono do ficheiro, geralmente a pessoa que o criou ou que foi indicada para o manter, este utilizador pode alterar todas as permissões do ficheiro. Existe a categoria “group” a que pertencem todos os utilizadores que estejam no mesmo grupo que o “owner” (a próxima secção explica melhor o conceito de grupo e hierarquia de utilizadores), esta categoria é essencial em projectos (empresariais ou universitários) que são desenvolvidos por uma equipa de colaboradores, uma vez que permite especificar permissões para um grupo bem definido de utilizadores, diferenciando-os do utilizador genérico e do dono do ficheiro. Finalmente, existe a categoria “all” na qual se incluem todos os utilizadores que não pertençam a nenhuma das categorias anteriores; tipicamente este conjunto de permissões será o mais restritivo, uma vez que não há nenhuma espécie de controlo sobre quem acede ao ficheiro, qualquer utilizador registado no sistema está automaticamente incluído nesta categoria. É de referir a existência de um super-utilizador (*root*) ao qual não é atribuído um conjunto de permissões em cada ficheiro, uma vez que se trata do administrador de sistema que, por essa razão, tem acesso total a todo e qualquer ficheiro.

Existem duas notações diferentes para traduzir as permissões de um ficheiro, uma de alto nível que é usada nas listagens longas de ficheiros (/comandols -l), e outra que consiste na representação binária (frequentemente traduzida para decimal) das permissões, tal como é feita pelo sistema operativo. Fazendo uma listagem detalhada do conteúdo de uma directoria, poderá ver qualquer coisa como:

```
drwx-----x 25 padilha webteam 1024 Sep 23 01:12 .
drwxr-xr-x 19 root root 1024 Sep 5 14:53 ..
-rw----- 1 padilha webteam 9917 Sep 22 19:45 .bash_history
-rw-r--r-- 1 padilha webteam 24 Jul 11 12:09 .bash_logout
-rw-r--r-- 1 padilha webteam 230 Jul 11 12:09 .bash_profile
-rw-r--r-- 1 padilha webteam 171 Sep 12 15:36 .bashrc
-rw-r--r-- 1 padilha webteam 154 Jul 11 12:09 .cshrc
drwxr-xr-x 5 padilha webteam 1024 Sep 18 08:26 .dia
drwxr-xr-x 5 padilha webteam 1024 Sep 12 11:24 GNUstep
drwx----- 2 padilha webteam 1024 Sep 15 19:26 Mail
-rw-r--r-- 1 padilha webteam 949 Sep 18 08:46 urls
lrwxrwxrwx 1 padilha webteam 17 Sep 6 00:09 webteam -> /mnt/hd2/webteam/
```

No início da linha, se se excluir o primeiro carácter que identifica o tipo de ficheiro, os restantes nove caracteres constituem três conjuntos de três atributos (um carácter por atributo). Como já foi dito antes, o primeiro conjunto é para o *owner*, o segundo para *group* e o terceiro para *all*. Nesta notação, a permissão de leitura representa-se por um “r” (do inglês *read*), a permissão de escrita por “w” (*write*) e a permissão de execução por “x” (*execute*). Assim, a sequência *-rwx-r-x-r* dá permissões totais ao *owner* do ficheiro (leitura, escrita e execução), permissões de leitura e execução a todos os utilizadores que pertençam ao mesmo grupo que o

dono do ficheiro, e apenas permissão de leitura a todos os utilizadores que não se enquadrem em nenhuma das categorias anteriores.

A segunda notação, de baixo nível, traduz a forma como o sistema operativo guarda as permissões do ficheiro, usando códigos binários de 3 *bits* para cada conjunto de atributos (1 *bit* por atributo). Dentro de cada conjunto de permissões, a permissão de execução ocupa o primeiro *bit* menos significativo, a permissão de escrita o segundo *bit* e a de leitura o terceiro e último *bit*. Normalmente, para facilitar a leitura destes códigos, opta-se pela representação decimal de cada um dos conjuntos. Usando o mesmo ficheiro que atrás referimos como tendo permissões *-rwx-r-x-r*, vamos ver de que forma podemos escrevê-las nesta notação. Sempre que uma dada permissão é concedida, o bit correspondente vai a 1, assim, para traduzir o conjunto de permissões do *owner*, será necessário colocar todos os três *bits* a 1, obtendo a sequência *111*, que convertida para decimal resulta em $4+2+1=7$. Seguindo o mesmo procedimento para o conjunto referente a *group*, será necessário colocar o primeiro e último *bit* a 1, mas deixar o bit de escrita em 0, obtendo assim a sequência *101*, que convertida para decimal resulta em $4+0+1=5$. Finalmente, para a categoria *all* só é necessário colocar o *bit* menos significativo a 1 e deixar os outros dois a 0, obtendo assim a sequência *001* que convertida para decimal resulta em $0+0+1=1$. Ou seja, dizer que um determinado ficheiro tem permissões *-rwx-r-x-r* ou *751* é exactamente a mesma coisa, com a vantagem da facilidade de leitura para o primeiro caso e de compactação no segundo. Esta segunda notação, por ser mais rápida de ler e escrever é, normalmente, preferida por utilizadores mais experientes, que não têm que pensar para traduzir o código binário (ou a sua representação decimal) para aquilo que ele representa.

Linking

Ajudando à implementação do conceito de sistema de ficheiros virtual, o Linux disponibiliza a possibilidade de manter referências/apontadores entre ficheiros e directorias. Usando o comando `ln` é possível criar dois tipos de *link* entre ficheiros:

- Um ***Hard-link*** é só um segundo nome para um ficheiro já existente; em termos técnicos, partilham o mesmo *inode*, e o *inode* contém toda a informação sobre o ficheiro—de facto, não é incorrecto dizer que o *inode* é o ficheiro. Os *hard-links* não podem ser criados entre directorias, nem entre sistemas de ficheiros diferentes.
- Um ***symbolic-link*** (ou simplesmente *symlink*), pelo contrário, é um tipo especial de ficheiro (que nem todos os *kernels* suportam) que se refere a outro ficheiro através do seu nome. A maior parte das operações realizadas sobre o ficheiro de ligação (abrir, ler, escrever, etc.), são desreferenciadas pelo *kernel* e aplicadas sobre o ficheiro original. A remoção, por exemplo, não é desreferenciada, operando directamente sobre o *link*. É possível criar *symlinks* entre directorias e através de sistemas de ficheiros distintos.

Para criar um *hard-link* deve escrever `ln [OPTION]... TARGET [LINKNAME]`, se não especificar o último parâmetro, o nome da ligação será o mesmo do ficheiro de destino. Para criar um *symlink* basta acrescentar a opção `-s`.

Esta possibilidade permite, por exemplo, ter vários servidores de X instalados, mantendo em `/etc/X11/X` um apontador para o servidor que se pretende usar no momento. Assim, se todos os programas e scripts se referirem a este link simbólico (*symbolic link*), basta alterar o local para onde ele aponta para passar a utilizar um servidor de X diferente, sem ter de reiniciar o computador, ou fazer qualquer outro tipo de reconfiguração.

Esta funcionalidade permite igualmente manter uma cópia única de uma dada informação, mas fazê-la estar presente em mais do que o sítio. Isto é, é possível que tanto `webmaster/pagina` como `/ficheiro/var/www` apontem a mesma informação que pode estar guardada numa terceira directoria. Este tipo de flexibilidade é extremamente útil quando coexistem dois ou mais sistemas operativos na mesma máquina (especialmente se um deles não conseguir ler a partição Linux) pois permite que as alterações feitas em determinada área (virtual!) da conta de um utilizador (que tipicamente estará na partição Linux) se reflectam directamente em ficheiros que se encontram fisicamente noutra partição e que, por qualquer motivo, têm que lá estar.

Ficheiros Escondidos

Tal como acontece noutros sistemas operativos, o Linux também permite a criação de ficheiros ocultos. Em vez de ser necessário atribuir a propriedade “hidden” a um ficheiro (como acontece, por exemplo, nos sistemas operativos da Microsoft), para definir um ficheiro escondido em Unix/Linux basta que o seu nome comece por um “.”. Por exemplo, renomear o ficheiro `hello.c` para `.hello.c` faz com que este passe a ser um ficheiro oculto, não aparecendo por isso na listagem de um `ls` normal.

Nota: Para ver todos os ficheiros de uma directoria, incluindo os ocultos, deverá usar-se o comando na forma `ls -a`.

Ficheiros Importantes

Existem alguns ficheiros de configuração importantes, que o utilizador comum deve ter o cuidado de configurar de acordo com as suas necessidades ou uso pessoal. Como se trata de ficheiros de configuração, que não são necessários no trabalho rotineiro do utilizador, são quase sempre ficheiros escondidos. Apresentamos de seguida alguns deles, mas o leitor deve estar consciente que existem muitos mais ficheiros deste tipo, eventualmente tão ou mais importantes, que não devem ser ignorados. Todos estes ficheiros devem estar colocados directamente na raiz da directoria pessoal do utilizador.

.login [c-shell]

Trata-se de um ficheiro de configuração oculto que é executado quando o utilizador entra na conta. Este ficheiro contém vários parâmetros importantes de configuração e como tal **nunca deve ser apagado!** Pode no entanto ser alterado, através de um simples editor de texto, mas atenção, não alterar nada, a não ser que saiba exactamente o que está a fazer!

Pode-se alterar o formato do *prompt* da nossa consola através da alteração do comando “set prompt=”. Por vezes ocorrem erros aquando da utilização do cliente de mail *pine*. Isto deve-se à configuração do terminal, que para o *pine* poder funcionar correctamente tem que ser do tipo “vt100”. Pode-se evitar este erro através da adição da seguinte linha no ficheiro:

```
set term=vt100
```

São de uma grande utilidade a criação de *aliases*, uma espécie de atalho para correr varios comandos sem grande esforço. Entende-se melhor com um exemplo:

- caso queira fazer uma listagem de todos ficheiros da minha conta, poderia usar o comando `ls` na forma `ls -la |more`, ora estar muitas vezes a escrever isto dá trabalho e origina erros. Podemos então definir no nosso `.login` uma nova linha com o texto seguinte `alias dir 'ls -la|more'`. Da próxima vez que voltar a entrar na conta basta digitar “dir” e temos a nossa listagem.
- para se ser questionado quando se apaga um ficheiro, usar o comando `rm` como *alias* de `rm -i`. Fazendo `alias rm 'rm -i'`.

Nota: podia-se executar o comando `source .login` e alteração feita no ficheiro fica imediatamente activa, ideal para testes.

.xinitrc e .xsession São dois ficheiros diferentes que fazem a mesma coisa: definir qual o gestor de janelas que deve ser utilizado numa sessão de X. Basta escrever o nome de um gestor de janelas na primeira linha, mas normalmente opta-se por escrever todos os gestores de janelas que se usam com mais frequência, um por linha, tendo o cuidado de comentar (com um `#` todas as linhas excepto a que contém o gestor de janelas que se pretende usar no momento. Fazer isto permite facilitar a escolha do gestor de janelas, uma vez que passa a ser necessário apenas comentar uma linha e descomentar outra para trocar de gestor.

Nota: O ficheiro `.xsession` é usado preferencialmente em sistemas GNU, como por exemplo Debian.

.forward Como o próprio nome indica, este ficheiro contém os endereços de email para onde todo o correio que chegue à conta do utilizador deve ser reencaminhado. Deverá ser escrito um endereço por linha, sem vírgulas ou qualquer outro

separador. Se preceder a primeira linha de uma `\`, o correio será também deixado na caixa de correio local. Se estiver a usar filtros de *mail*, como o *procmil* por exemplo, pode ser necessário fazer algumas alterações a este ficheiro. Como essas alterações dependem do filtro que se está a usar, não vamos elaborar mais sobre o assunto.

.signature Este é um ficheiro próprio do PINE, embora também seja frequentemente utilizado por outras aplicações. Serve para guardar uma “assinatura” que será acrescentada a todas as mensagens de email que enviar a partir do PINE. Não é necessário usar qualquer tipo de comandos, basta escrever o texto tal e qual como se deseja que apareça na mensagem.

2.2.3 X Windows

Introdução

Tradicionalmente, o Unix associa-se a uma série de comandos com nomes difíceis de lembrar que são usados num terminal de texto. Essa foi a forma tradicional como o Unix foi desenvolvido, e continua a ser o modo de operação mais rápido e preferido por pessoas com muita experiência, mas existe também o modo gráfico usando o X que resulta mais fácil e agradável de usar para quem se está a iniciar no sistema operativo.

Do X Windows convém salientar o facto de estar estruturado numa arquitectura servidor/cliente, o que significa que o gestor de janelas que o utilizador vê, não é mais do que um **cliente** que se liga a um **servidor** de X, este completamente transparente para o utilizador. A grande vantagem deste sistema é permitir que o utilizador escolha, em qualquer momento, o gestor de janelas da sua preferência, sem ter que fazer nenhuma configuração adicional.

Num ambiente gráfico como o X são disponibilizadas áreas de trabalho virtuais. A gestão dessas áreas de trabalho está a cargo do gestor de janelas. Normalmente as várias áreas de trabalho são configuráveis independentemente, podendo cada uma, por exemplo, apresentar uma imagem de fundo diferente.

Uma outra peculiaridade do sistema de janelas X, para quem não esteja familiarizado com ele, é que os três botões do rato (esquerdo, central e direito) são importantes em muitas aplicações. Quando o rato tem apenas dois botões, normalmente «emula-se» um rato com três botões, sendo o botão do meio obtido pela pressão simultânea dos dois botões.

Servidor/Cliente

O servidor de X foi construído com base numa arquitectura cliente/servidor que é um dos conceitos mais avançados na arquitectura do X. Isto permite um funcionamento completamente transparente através de uma rede de computadores e permite,

por exemplo, correr um programa num computador e visualizar o seu *output* gráfico num qualquer ecrã de outro computador.

Esta funcionalidade é conseguida pois o servidor é o programa que controla directamente os dispositivos do *hardware* e as aplicações são os clientes. Estas aplicações apenas mandam mensagens muito simples (exemplo:criar janela, mover janela, desenhar linha, desenhahr texto...) ao servidor para que este execute as opeações desejadas.

O servidor X tambem tem a tarefa de monitorizar os dispositivos de entrada e detectar novos eventos gerados pelo utilizador, tais como o mexer do rato, premir uma tecla etc..

Em termos de tempos de execução esta abordagem pode provocar ligeiras perdas de performance em face aos sistemas monolíticos, no entanto este efeito não é relevante para a esmagadora maioria das aplicações pois estas passam 99.9% do tempo a espera do *input* do utilizador.

KDE

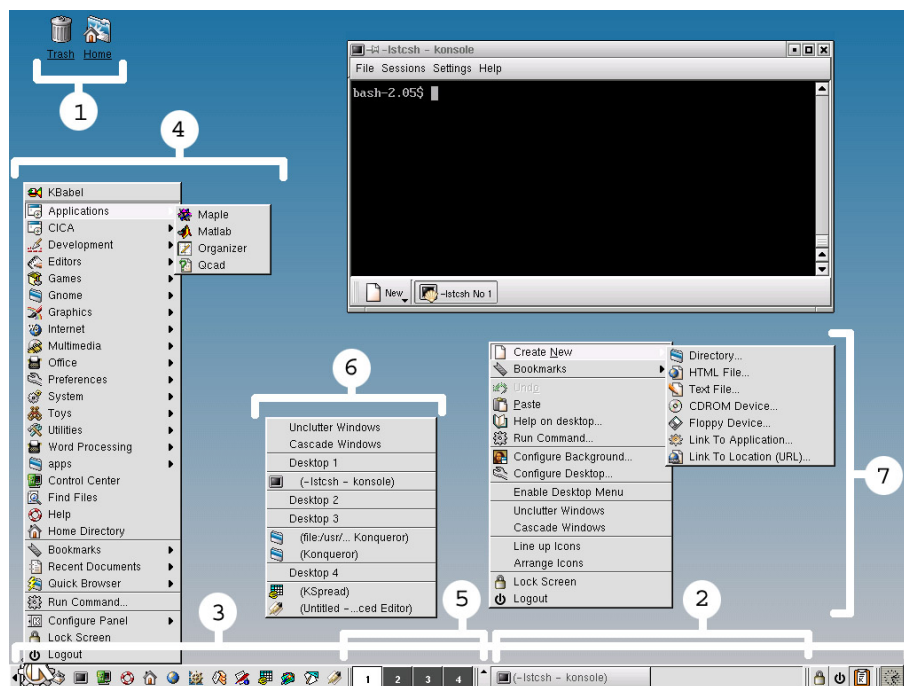


Figura 2.27: Ambiente de trabalho no KDE.

Descrição

KDE é o gestor de janelas mais *user-friendly* disponível em Linux. É o ideal para pessoas habituadas a trabalhar em Windows 9x/NT devido à sua facilidade de utili-

zação e semelhança a nível gráfico. O interface é bastante intuitivo permitindo que pessoas novas ao ambiente trabalhem com eficiência e rapidez.

Interface

Possui um *desktop* onde se podem colocar atalhos (1) para aplicações ou *drives*, uma *taskbar* (2) para comutar facilmente entre as aplicações que estão a correr, e uma barra de ferramentas (3) para executar as principais aplicações mais rapidamente. Todos os programas podem ser corridos através do menu de aplicações (4) acedido no canto inferior esquerdo na barra de ferramentas. Este gestor de janelas tem ainda a vantagem de trazer de raiz uma grande quantidade de aplicações: calculadora, editor de texto, jogos vários, editor de imagens, explorador de fractais, compressor de ficheiros, etc (que podem ser usados também em qualquer outro gestor de janelas diferente do KDE).

Por omissão existem quatro áreas virtuais de trabalho podendo este número ser estendido até oito. A transição entre os vários ambientes virtuais é feita através de botões na barra de aplicações (5). Cada ambiente de trabalho pode ser configurado independentemente com diferentes imagens de fundo, ícones e até nomes.

O botão central e o botão direito do rato permitem-nos aceder a diferentes menus. Pressionando o central temos acesso a um menu (6) com as aplicações a correr nos vários ambientes de trabalho. O botão direito, por sua vez, dá-nos acesso a um menu de contexto (7) com opções específicas relativas à área onde se encontra o cursor do rato nesse momento.

Configuração

A configuração do KDE é muito fácil e totalmente controlada através de uma única aplicação denominada «KDE Control Center». Aqui pode-se alterar todos os aspectos do gestor de janelas como a imagem de fundo, o número de ambientes de trabalho e os seus nomes, a língua utilizada, o gestor de ficheiros, etc. A navegação entre as várias opções é fácil e rápida devido ao seu esquema em árvore (1) e ao uso de separadores.

2.2.4 Consola

O que é uma “consola”?

Uma consola, no sentido que nos interessa, é constituída por um dispositivo de entrada (o teclado), um dispositivo de saída (o monitor) e “algo” que trabalha sobre a entrada para produzir a saída (a *shell*). Este é o conceito básico de consola. Existem outros tipos de consolas, como as ditas consolas gráficas, mas quando se fala de interface de consola é a estas consolas (também ditas consolas de texto) que se está a referir.

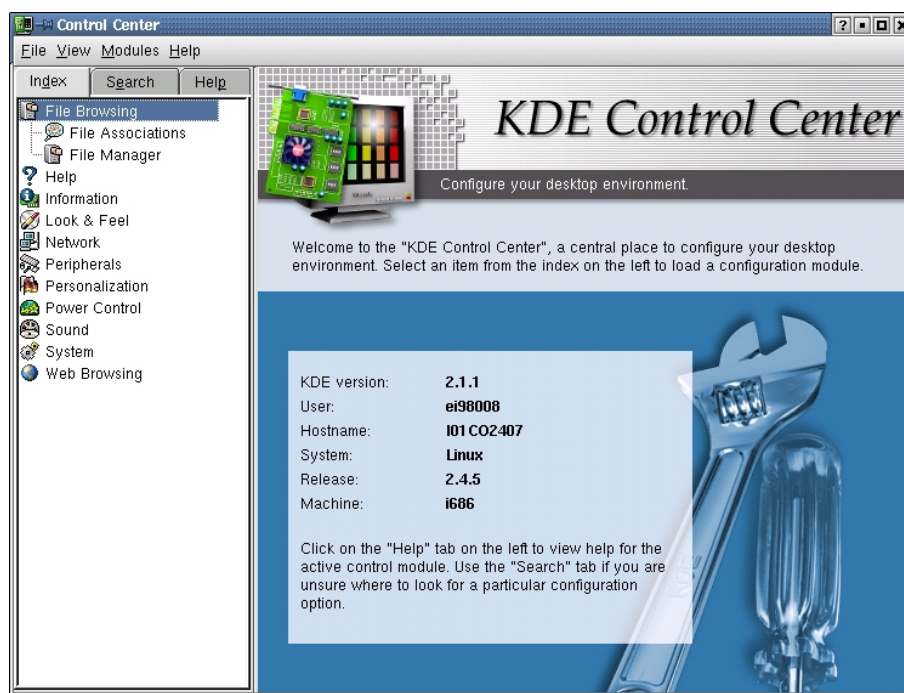


Figura 2.28: KDE Control Center.

O conceito vem dos primeiros tempos do Unix em que existia um computador central ao qual se ligavam vários terminais através de várias portas série. Um terminal possuía apenas um teclado e um monitor, sendo todo o trabalho de processamento realizado no computador central (onde corria a *shell*). Embora este tipo de interface seja bastante limitada em termos visuais, é característica dos sistemas Unix/Linux, sendo a sua utilização quase indispensável.

Características de uma “shell”

Standard Input e Standard Output Muitos comandos de Linux recebem dados através do chamado *standard input* e enviam os resultados para o *standard output* (frequentemente abreviados por *stdin* e *stdout*). A *shell* arranja as coisas de modo a que o *standard input* seja o teclado e o *standard output* seja o ecrã.

Aqui está um exemplo usando o comando `cat`. Normalmente, o comando `cat` lê os dados de todos os ficheiros especificados na linha de comandos, e envia os resultados directamente para o `stdout`. Assim, usando o comando

```
/home/larry/papers# cat history-final masters-thesis
```

mostra o conteúdo do ficheiro `history-final` seguido do conteúdo de `master-thesis`.

Contudo, se não especificar um ficheiro, o `cat` vai ler os dados do `stdin` e enviá-los de volta para o `stdout`. Aqui está um exemplo:

```
/home/larry/papers# cat
Hello there.
Hello there.
Bye.
Bye.
Ctrl-D
/home/larry/papers#
```

Cada linha que escrever é imediatamente devolvida pelo `cat`. Quando se lê do *standard input* indica-se que a entrada de dados “terminou” enviando um sinal EOT (end-of-text - final do texto), que geralmente pode ser feito carregando em Ctrl-D.

Aqui está outro exemplo. O comando `sort` lê linhas de texto (também do *stdin*, a não ser que se especifique o nome de pelo menos um ficheiro) e envia o resultado ordenado para o *stdout*. Experimente fazer o seguinte:

```
/home/larry/papers# sort
bananas
cenouras
alperces
Ctrl-D
alperces
bananas
cenouras
/home/larry/papers#
```

Assim é possível ordenar a lista de compras por ordem alfabética... não é o Linux útil?

Redirecionamento Agora, vamos imaginar que pretende enviar os resultados do `sort` para um ficheiro, para gravar a lista de compras no disco. A *shell* permite-lhe redireccionar o *standard output* para um ficheiro, usando o símbolo “>”. Aqui está um exemplo do seu funcionamento:

```
/home/larry/papers# sort > shopping-list
bananas
cenouras
alperces
Ctrl-D
/home/larry/papers#
```

Como pode ver, o resultado do comando `sort` não é mostrado, mas em vez disso é guardado no ficheiro chamado `shopping-list`. Vamos ver o que contém o ficheiro:

```
/home/larry/papers# cat shopping-list
alperces
bananas
cenouras
/home/larry/papers#
```

Agora já pode não só ordenar a sua lista de compras, como também guardá-la! Mas vamos supôr que tem a lista de compras original, não ordenada, guardada no ficheiro `items`. Uma forma de ordenar a informação e guardá-la num ficheiro seria passar como argumento do comando `sort` o nome do ficheiro a lêr em vez do *standard input*, e redireccionar os resultados como já foi feito anteriormente, por exemplo desta forma:

```
/home/larry/papers# sort items > shopping-list
/home/larry/papers# cat shopping-list
alperces
bananas
cenouras
/home/larry/papers#
```

Existe, contudo, outra forma de fazer isto. É possível redireccionar não só o *standard output*, como também o *standard input*, usando o símbolo “<”.

```
/home/larry/papers# sort < items
alperces
bananas
cenouras
/home/larry/papers#
```

Tecnicamente, `sort < items` é equivalente a `sort items`, mas este exemplo permite ilustrar o seguinte ponto: `sort < items` comporta-se como se a informação do ficheiro `items` fosse escrita directamente para o *standard input*. A *shell* trata do redireccionamento. O comando `sort` não recebeu como parâmetro o nome do ficheiro (`items`) para ler e tanto quanto diz respeito ao comando `sort`, a informação continua a vir do *standard input*, tal e qual como se estivesse a escrever o conteúdo do ficheiro directamente no seu teclado.

Comandos de sistema

Para quem já trabalhou com DOS, os comandos de sistema do Linux são bastante fáceis de aprender. Existe, inclusivamente, um *How-To* dedicado a esses casos. Poderá encontrá-lo em `/usr/doc/HOWTO/dos2linux` em qualquer sistema Linux.

man Apesar de não poder ser considerado um comando j de sistema, o man é de extrema importância tanto para o utilizador inexperiente como para o administrador de sistema. Dada a sua importância para a compreensão dos comandos que se seguem, optámos por o inserir nesta secção.

O man permite ver a *manpage* de comandos, funções e ficheiros especiais. Uma *manpage* é um documento que explica o objectivo, campo de aplicação e funcionamento do ficheiro, comando ou função, contendo frequentemente exemplos de utilização. Sempre que se tem dúvidas sobre o funcionamento de alguma coisa em Unix/Linux é de boa prática verificar se existe uma *manpage* associada. Como exercício propomos que o leitor consulte a *manpage* do próprio comando man, escrevendo `man man`.

ls O comando ls faz uma listagem dos ficheiros existentes no directório onde se encontra, ou então de uma outra directoria, caso seja especificada. É possível passar alguns argumentos, refinando a listagem que se obtém. Como exemplo temos `ls -la`, que mostra o conteúdo total (permissões e ficheiros ocultos inclusivé) da directoria actual.

cd Este comando permite a mudança de directoria. O caminho especificado pode ser absoluto ou relativo, isto é, pode ser o caminho completo da directoria ou apenas o seu nome. Se apenas se escrever `cd` muda-se para a directoria \$HOME (directoria pessoal) do utilizador.

Alguns exemplos:

- `cd /` - passa para a raiz de todos os directorias (`root`);
- `cd ..` - sobe uma directoria na árvore;
- `cd ../..` - sobe duas directorias na árvore;
- `cd /usr/home/` - muda para `/usr/home`;

cp Permite copiar ficheiros. Usa-se normalmente na forma `cp ficheiro1 ficheiro2`. Aceita também caminhos completos e/ou relativos dos ficheiros. Algumas opções úteis deste comando são:

- `-i` pede confirmação antes de efectuar a cópia
- `-R` copia uma directoria recursivamente (todo o seu conteúdo, incluindo sub-directorias).

Exemplo: Para copiar o ficheiro `teste.txt` localizado na directoria pessoal do utilizador para a directoria `/usr/tmp`, poderá fazer-se:

- `cp teste.txt /usr/tmp/` - escrito desta forma, é preciso que a directoria actual seja a directoria pessoal do utilizador;
- `cp /teste.txt /usr/tmp/` - escrito desta forma, funciona correctamente qualquer que seja a directoria actual.

mv Permite mover ficheiros. No entanto a sua função mais útil é renomear um ficheiro, isto acontece quando se move o ficheiro de uma directoria para a mesma directoria. Por exemplo, para renomear o ficheiro `antigo.txt` para `novo.txt` usar `mv antigo.txt novo.txt`. Este comando aceita também caminhos completos e/ou relativos dos ficheiros. Uma opção útil deste comando é:

- `-i` - pede confirmação antes de efectuar a cópia;

rm Este comando permite remover (apagar) ficheiros e directorias. As directorias apenas serão removidas se estiverem vazias, ou então caso se use as opções `-rf`. A opção `r` (*recursive*) faz com que sejam apagados todos os ficheiros e subdirectorias dentro da directoria especificada. A opção `f` faz com que não se seja questionado do que se está a apagar. Recomenda-se alguma atenção no uso destas opções, uma vez que não é possível recuperar os ficheiros apagados! Por esta razão recomenda-se a utilização da opção `-i` que questiona o utilizador antes de apagar um ficheiro.

Alguns Exemplos:

- `rm -i teste.txt` - Apagar o ficheiro `teste.txt`. A *shell* pergunta `Remove teste.txt?`. Carregar em `y` ou escrever `yes`
- `rm *` - Apagar todos os ficheiros da directoria actual.

mkdir Diminutivo de *make directory*, é usado para a criação de novas directorias. Tal como outros comandos suporta caminhos absolutos ou relativos. **Exemplo:**

- `mkdir public_html` - Criar o directório `public_html` na directoria actual.
- `mkdir /usr/tmp/ei01000` - Criar directoria `ei01000` em `/usr/tmp/`

rmdir Diminutivo de *remove directory*, é usado para apagar directorias. No entanto, estas apenas serão apagadas se estiverem vazias. O utilizador só pode apagar as directorias para as quais tem permissões de escrita. A sua directoria pessoal (`$HOME`) é uma delas. Deve no entanto ter muita atenção às directorias que apaga uma vez que muitas são vitais para o funcionamento de programas! Tal como outros comandos, aceita caminhos completos e/ou relativos para a criação das directorias. **Exemplo:**

- Para apagar o directório "fotos" que tem "foto1.jpg" e "foto2.jpg".
- `rm foto1.jpg`
- `rm foto2.jpg`
- `cd ..`
- `rm fotos`

pwd Diminutivo de *print working directory*. Indica o directório em que o utilizador se encontra actualmente. Este comando torna-se bastante útil quando estamos a navegar pela hierarquia dos directorias.

Exemplo:

- `pwd` - Ver a directoria actual.

chmod Este comando permite alterar as permissões de ficheiros ou directorias. Na forma mais simples, aceita dois argumentos: em primeiro lugar as permissões que se deseja alterar (usando a notação extensa) ou restabelecer (usando a notação compacta) e em segundo lugar o nome do ficheiro ou directoria (podem usar-se *wildcards*³). A melhor forma de explicar o funcionamento do comando é usar alguns exemplos:

chmod a+r * - dá permissão de leitura à categoria *all* de todos os ficheiros da directoria actual;

chmod g-rwx hello.c - remove todas as permissões à categoria *group* do ficheiro `hello.c`;

chmod 700 . - atribui permissões 700 à directoria actual, ou seja, permissões totais para o *owner* e nenhuma permissão para qualquer outro utilizador;

chmod 755 mail.cgi - atribui permissões 755 ao ficheiro `mail.cgi`, ou seja, permissões totais para o *owner* e permissões de leitura e execução para todos os outros utilizadores;

Como se pode ver, usando a notação extensa **alteram-se** (adiciona-se, remove-se...) as permissões de um ficheiro, conjunto de ficheiros ou directoria, enquanto que com a notação compacta **estabelecem-se** essas mesmas permissões (não importa quais eram anteriormente).

³Caracteres que substituem uma ou mais letras no nome de um ficheiro ou directoria. O * substitui várias letras e o ? apenas uma. Por exemplo `fotos*` refere-se aos ficheiros `foto1`, `fotoViagem`, `foto2`, etc. Enquanto `foto?` refere-se apenas aos ficheiros `foto1` e `foto2` não referindo `fotoViagem`.

chown Este comando permite alterar o *owner* de um ficheiro ou directoria. Só o dono actual do ficheiro ou o super-utilizador o pode fazer com sucesso. A sintaxe do comando é: `chown <nome-do-novo-dono> <nome-do-ficheiro>`.

kill Este comando permite enviar mensagens pré-estabelecidas ou sinais aos processos. Os sinais mais utilizados são:

- HUP Termina e reinicializa o processo
- TERM Termina o processo.
- KILL Força a terminação do processo. Usar só em último recurso porque pode deixar recursos do sistema ocupados.

Experimente executar uma shell e escrever na linha de comando `top`. Aparece uma lista dos processos que estão a correr na máquina. Procure na coluna mais à direita o processo `top` e aponte o seu PID correspondente que está na primeira coluna da mesma linha. Abra outra shell e escreva `kill -s TERM PID` (substituir onde está escrito PID⁴ pelo número do processo anterior). Irá verificar que na outra shell o comando `top` parou.

ps Este comando permite ver os processos a correr na máquina. É uma ferramenta vital durante a execução de uma sessão de trabalho, pois permite ver não só o nome dos processos como também informações específicas sobre a sua execução (nomeadamente a memória usada, o tempo de CPU, a percentagem de CPU usada, etc). Tal como os outros comandos Unix/Linux é bastante configurável tendo um vasto leque de opções à disposição do utilizador. As opções mais usadas são `l` para obter o máximo de informações possível sobre cada processo, `x` para obter a listagem de processos a correr no modo gráfico e `a` para obter a listagem de processos de outros utilizadores. Estas opções podem ser utilizadas ao mesmo tempo para configurar a listagem de saída do `ps`. Por exemplo, `ps lx` permite ver os processos a correr em modo gráfico, mas apenas os que pertençam ao utilizador actual.

top Este comando permite obter uma listagem dos processos que estão a ocupar mais recursos na máquina. De cada processo listado é mostrado, entre outros, o nome, PID, espaço ocupado em memória, percentagem de CPU ocupada, e o estado (activo, dormente, etc.). Para além de realizar a listagem, esta aplicação também permite fazer a gestão de processos, de onde se destaca a possibilidade de executar um `kill` pressionando a tecla `[k]` e inserindo de seguida o PID do processo e o sinal com que se pretende matar o processo.

⁴Em Linux todos os processos a correr no sistema tem um número que o identifica - PID (*Process ID*).

Aplicações Importantes

talk O Unix/Linux possui um sistema simples de comunicação entre utilizadores que estejam ligados em rede. Apesar de ser em consola a sua interface é bastante intuitiva e parecida com o tradicional IRC. O ecrã é dividido a meio (uma metade pertence ao utilizador que iniciou a ligação e outra metade ao utilizador de destino), bastando começar a escrever logo que a ligação seja feita. Para efectuar uma ligação é preciso escrever o comando `talk <login do utilizador destino>`, estabelecendo-se a ligação assim que o utilizador de destino responder com `talk <login do utilizador origem>`. Para um utilizador ser notificado de tentativas de comunicação tem que ter o ambiente de trabalho configurado para aceitar mensagens. Para tal usa-se o comando `mesg`. Recebe como argumento as opções `y` ou `n`, sendo a primeira para aceitar comunicações e a segunda para não aceitar. Se não se passar nenhum argumento ao `mesg` este indica a configuração actual.

pico O `pico` é um editor de texto de consola desenvolvido pela Universidade de Washington, que acompanha o seu cliente de mail (`pine`).

É um programa de fácil aprendizagem, uma vez que os comandos principais estão indicados nas duas últimas linhas do ecrã. Apenas a título de exemplo, `CTRL+X` permite sair do programa, `CTRL+O` guarda o ficheiro actual, `CTRL+K` faz “cut” de linhas completas e `CTRL+U` faz “uncut” das últimas linhas. `CTRL+W` permite encontrar uma palavra no texto e `CTRL+R` acrescenta um ficheiro ao ficheiro aberto, a partir da posição actual do cursor (“inlining”).

Esta aplicação é normalmente substituída em sistemas GNU (como por exemplo *Debian*) por um seu clone, o `nano`. Para mais informação, o ficheiro de ajuda do PICO pode ser consultado online em <http://www.msoe.edu/tritt/cs185/picohelp.html>

pine O `pine` é, provavelmente, o cliente de mail de consola mais usado. Para além de receber e enviar correio, também é possível guardar mensagens em várias directorias, criar, gerir e usar livros de endereços, definir uma “assinatura” para acrescentar a todas, ou só algumas, mensagens enviadas, configurar filtros, etc.

Esta aplicação é da autoria da Universidade de Washington e possui um interface muito semelhante ao do PICO, apesar de ser muito mais complexo. Talvez por possuir um interface muito intuitivo, necessitar de relativamente pouca ou nenhuma configuração e consumir poucos recursos, o PINE é quase um *standard*, estando presente em praticamente todos os sistemas Unix/Linux.

É possível criar filtros para o PINE, mas trata-se de um assunto avançado, para o qual existem tutoriais e *how-to's* bastante bons. Sugerimos consultar a página de manual do PINE para obter mais informação, ou directamente a página dedicada ao PINE da Universidade de Washington (em <http://www.washington.edu/pine/>).

LPR Este comando permite imprimir documentos. O comando a usar é `lpr <filename>`, onde `filename` é o nome do ficheiro a imprimir. Este comando

também permite especificar funcionalidades adicionais na impressão. Por exemplo:

- `lpr -P <printer> <filename>` No caso de haver várias impressoras disponíveis, utilizar a impressora chamada `printer`
- `lpr -o sides=two-sided-long-edge filename` Imprimir dos dois lados da folha
- `lp -o media=A4 -o landscape filename` Imprimir em folhas A4 e no formato `landscape`
- `lpr -o page-ranges=1-4,7,9-12 filename` Imprimir das páginas 1 a 4, 7 e da 9 à 12

LPQ Comando que permite ver a fila de espera de documentos a imprimir numa determinada impressora. Experimente executar o comando `lpq`.

telnet O `Telnet` é um serviço de rede que permite fazer *login* num terminal remoto, tendo o utilizador acesso à máquina exactamente da mesma forma que teria se estivesse fisicamente em frente do computador.

Este serviço está estruturado em servidor/cliente, sendo por isso necessário que a máquina anfitriã esteja a correr um servidor de `telnet` para que a aplicação cliente local consiga estabelecer uma sessão.

O serviço de `telnet` permite um acesso fácil e bastante completo a uma shell Unix/Linux remota, podendo assim o utilizador aceder à sua área de trabalho seja qual for o sítio e plataforma em que se encontra (quase todos os sistemas operativos possuem pelo menos um cliente de `telnet`).

ssh O `ssh` permite estabelecer uma ligação semelhante ao `telnet` entre duas máquinas através de um canal encriptado (o algoritmo utilizado na encriptação pode ser configurado, mas é geralmente o 3DES). Ao contrário do que acontece com o `telnet`, o `ssh` assume que o `login` do utilizador na máquina remota será o mesmo utilizado na máquina local, pelo que será necessário acrescentar a opção `-l <username-remoto>` se tal não acontecer. É sempre preferível utilizar o `ssh` em desfavor do `telnet`, pois é muito fácil “sniffar”⁵ as passwords, bem como toda a restante informação transmitida, quando a comunicação não é encriptada.

ftp Programa usado para transferir ficheiros entre computadores. O seu funcionamento é equivalente a uma shell mas com seus comandos específicos. Experimente escrever numa shell o comando `ftp gnomo.fe.up.pt`, para ter acesso ao servidor `gnomo` da FEUP. Aparecerá o seguinte texto:

⁵um “sniffer” é um software que coloca a placa de rede em modo promíscuo (a placa passa a receber todos os pacotes que circulam na rede, incluindo aqueles que não lhe são destinados) e faz captura e processamento dos pacotes recebidos.

```
Connected to gnomo.fe.up.pt.  
220 ProFTPD 1.2.2rc3 Server (ProFTPD Default Installation) [gnomo.fe.  
Name :
```

Deverá escrever o seu login e password para ter permissão de acesso aos seus ficheiros. Aparecerá o seguinte texto: ftp>

Os comando mais usados são:

- LS Visualiza os ficheiros existentes no servidor remoto
- !DIR Visualiza os ficheiros no computador local
- CD <nome directoria> Muda para a directoria <nome directoria> no servidor remoto
- LCD <nome directoria> Muda para a directoria <nome directoria> no computador local
- GET <nome ficheiro> Copia o ficheiro nome ficheiro do servidor remoto para o computador local
- PUT <nome ficheiro> Copia o ficheiro nome ficheiro do computador local para servidor remoto.
- HELP <comando> Informação sobre o comando comando. Se não se especificar o comando, dá uma lista de todos os comandos disponíveis.
- QUIT Sai do programa

sftp É um programa equivalente ao ftp mas que transfere dados de forma encriptada com o recurso ao ssh. Experimente executar uma shell e escrever na linha de comando sftp <login>@gnomo.fe.up.pt (substituir login pelo seu). Aparecá os seguinte texto no ecrã:

```
Connecting to gnomo.fe.up.pt...  
login@gnomo.fe.up.pt's password:
```

Deverá escrever o seu login e password para ter permissão de acesso aos seus ficheiros. Aparecerá o seguinte texto: sftp>

Alguns comandos disponíveis:

- HELP Ver todos os comandos reconhecidos e um sumário do que fazem.
- EXIT Sai do sftp para a shell
- LS Visualiza os ficheiros existentes no servidor remoto

- LLS Visualiza os ficheiros existentes no computador local
- GET <nome ficheiro> Copia o ficheiro nome ficheiro do servidor remoto para o computador local
- PUT <nome ficheiro> Copia o ficheiro nome ficheiro do computador local para servidor remoto.

2.3 TCPGATE

O Tcpgate é um serviço misto fornecido pelo CICA para a FEUP. Por um lado serve de *firewall*, escondendo a configuração interna da rede e simultaneamente obrigando à autenticação de todo e qualquer utilizador que, encontrando-se num ponto exterior à rede da FEUP, nela queira trabalhar. Por outro lado, sob o nome de VPN, presta o serviço de ISP para qualquer elemento da FEUP (professores, alunos, etc.) que o solicite, ao custo de uma chamada local.

Um utilizador que se encontre em casa, e por isso fora da rede interna da FEUP, terá obrigatoriamente que se autenticar no Tcpgate para ter acesso à sua área de trabalho. Supondo que um utilizador hipotético pretende aceder à máquina *gnomo*, o que teria que fazer seria:

Iniciar uma sessão de telnet para tcpgate.fe.up.pt Para fazer isto basta executar o comando `telnet tcpgate.fe.up.pt` e introduzir o login e password apropriados. Este login é específico do tcpgate e autentica o utilizador como membro autorizado da rede da FEUP. Depois de concluído a fase de login o melhor que tem a fazer é executar o comando `help` para ver quais as possibilidades que lhe restam. É de notar que o utilizador não se encontra ainda na sua conta, não tendo por isso acesso a nenhum dos seus ficheiros. Entre os comandos permitidos está o comando `telnet`.

Iniciar uma sessão no gnomo Escrevendo `telnet gnomo` (notar que deixa de ser necessário especificar o domínio - fe.up.pt - em que se encontra a máquina) estará a iniciar uma sessão de login na máquina gnomo. Basta introduzir o login e password do sistema Unix/Linux que lhe foi atribuída pelo CICA e o nosso utilizador imaginário terá acesso fácil a todos os ficheiros.

Outra alternativa as ligações por telnet ao tcpgate poderíamos fazê-lo usando comunicações encriptadas (usando ssh). E além de ser possível aceder à máquina tcpgate.fe.up.pt através de ssh, é possível estabelecer uma ligação encriptada a partir de lá.

2.4 Página Pessoal

É possível manter uma página pessoal alojada na FEUP, usando para isso a conta de Unix fornecida pelo CICA. O endereço é do tipo `http://www.fe.up.pt/~login` em que *login* é o *login* atribuído pelo CICA. Para criar uma página pessoal é necessário criar a directoria `public_html` na raiz da sua conta Unix, isto tanto pode ser feito em ambiente Windows (sendo a dita directoria criada na raiz do *drive Z:*), como através de uma *shell* Unix. No entanto, para que a página possa ser acedida por qualquer utilizador é necessário atribuir as permissões correctas, e estas permissões só podem ser alteradas usando uma *shell*. Anteriormente nesta sebeta existe uma secção inteiramente dedicada a permissões de ficheiros e formas de alterá-las, pelo que basta dizer que tanto a vossa directoria pessoal como a directoria `public_html` devem ter permissões 701 e todos os ficheiros dentro de `public_html` devem ter no mínimo 744 (alguns tipos de ficheiros podem precisar de permissões mais alargadas). Para que a página possa ser vista, é necessário que exista na directoria referida o ficheiro `index.html` ou `welcome.html`. Esta será a página inicial que aparece quando se digita o endereço.

Pode-se também criar páginas baseadas em php e scripts ou cgi's compilados, no entanto é necessário pedir autorização ao CICA para correr os scripts ou cgi's. Deverá solicitá-la para `csys@fe.up.pt`.

Após ter sido autorizado deve criar uma subdirectoria de `public_html` chamada `.cgi-bin` com permissões 701 através dos seguintes comandos:

```
mkdir /public_html/.cgi-bin
chmod 701 /public_html/.cgi-bin
```

Será nesta directoria que terá de colocar os seus scripts com permissão de leitura e execução para quem visita a sua página. Execute os seguintes comandos para tal:

```
cd public_html/.cgi-bin
chmod 755 *
```

O endereço dos cgi's é: `http://máquina.fe.up.pt/~login/htbin/nome.cgi` em que "máquina" deverá ser substituído pelo nome do servidor onde lhe forem atribuídas as permissões para correr cgi's e "nome.cgi" deverá ser substituído pelo cgi pretendido. Poderá parecer estranho invocar um CGI numa directoria que, em princípio, não existe `htbin`, contudo o sistema encarrega-se de transformar `htbin` em `public_html/.cgi-bin` o que significa que os programas ou *scripts* serão efectivamente acedidos na directoria com o nome `.cgi-bin`.

Por vezes quando se tenta aceder a uma página web recebe-se a seguinte mensagem:

```
Not Found The requested URL < página > was not found on this server.
```

Isto pode acontecer porque a página não existe, ou não tem permissões de acesso para a poder visualizar ou ainda devido a diferenças ao nível do nome e extensões dos ficheiros. Por exemplo, invocar-se `Index.htm` e o verdadeiro nome da página ser `index.htm` ou invocar-se `pagina.htm` e o verdadeiro nome da página ser `pagina.html`. É necessário ter atenção aos nomes dos ficheiros, especialmente o uso de maiúsculas e minúsculas, pois os servidores onde estão alojadas as páginas são *case sensitive* (as letras maiúsculas e minúsculas representam entidades diferentes).

Por fim ficam alguns links de sites sobre construção de páginas web, esperamos que sejam uteis:

- www.w3schools.com
- www.javascripts.com
- www.freepperlcode.com
- www.cidadela.org
- www.truquesedicas.com
- www.webmonkey.com
- www.php.net

2.5 GAUTI

O GAUTI é o Gabinete de Apoio à Utilização das Tecnologias de Informação.

O ensino, a investigação e a prestação de serviços ao exterior requerem uma crescente integração das tecnologias de informação e comunicação multimédia, que contribuem decisivamente para a excelência das actividades da FEUP e o seu posicionamento como escola de referência.

O GAUTI é a unidade responsável por manter uma estratégia para o uso das tecnologias da informação e comunicação nas actividades de ensino, aprendizagem, investigação e extensão universitária da FEUP, e tem por principal objectivo a prestação de serviços de apoio à utilização destas tecnologias a toda a comunidade académica da FEUP.

O GAUTI disponibiliza recursos informáticos, multimédia e audiovisuais e presta serviços de reprografia e de produção multimédia, podendo ser consultadas em detalhe as suas actividades em <http://www.fe.up.pt/gauti>.

Os novos alunos podem utilizar todos estes serviços, nas condições dos respectivos regulamentos de funcionamento, e podem ainda participar no desenvolvimento experimental de novos produtos e serviços ou nas equipas de apoio técnico ao auditório da FEUP.

Nota: Este texto foi gentilmente cedido pelo Director Executivo do GAUTI.