Parte I

UNIX

System V

SUMÁRIO

PARTE II
INTERNET

1. APRESENTAÇÃO

Esta apostila foi desenvolvida pela Facens para usuários leigos em UNIX e está em fase de aprimoramento . Desta forma, os autores estão aceitando sugestões e comentários para o aprimoramento deste documento.

2 INTRODUÇÃO

Bem-vindo ao UNIX. Se você não estiver familiarizado com computadores, o UNIX é um sistema operacional, ou seja, um conjunto de programas responsável por atividades como: realizar interface entre o hardware e os programas dos usuários, criando um ambiente mais adequado ao desenvolvimento de aplicativos; gerenciar os recursos de hardware existentes. Nos computadores do laboratório de informática da UFPR, encontramos o UNIX da HP-UX System V, no qual nos baseamos para concretizar esta apostila. Porém, existem outros UNIX, como o XENIX da IBM, o BSD da Berkeley, o Linux para PC, entre outros. O comando **man**, que é um help on-line, pode auxiliá-lo a trabalhar nestes UNIX's.

Lendo esta apostila, você encontrará muitos exemplos que ilustram com detalhes e os conceitos em cada capítulo. Na parte I, serão abordados conceitos e comandos básicos do sistema operacional UNIX. Na parte II você encontrará programas mais utilizados na rede Internet.

Podemos citar algumas características do UNIX:

- Capacidade multitarefas;
- Capacidade multiusuários;
- Portabilidade;
- Ampla seleção de programas;
- Comunicação e correio eletrônico;
- Biblioteca de software aplicativos.

3 COMO ENTRAR NO SISTEMA

No Sistema UNIX cada usuário possui uma identificação (que é chamada de *login* do usuário) e uma senha que permite o acesso exclusivo para as suas respectivas áreas. Esta senha é atribuída para o usuário no momento em que ele é cadastrado no sistema. Pode ser alterada sempre que o usuário sinta necessidade, e deve ser guardada segura e sigilosamente.

Sempre que se deseje entrar no sistema, deve-se digitar seu login e em seguida a sua senha. Caso não combinem será pedido para que se digite novamente os dados. Caso se obtenha sucesso na operação de login, o sistema executará algumas operações de inicialização (como ler os arquivos de configuração) e em seguida mostrará o "prompt" dos comandos, mostrando que o sistema pode ser usado.

4 MANIPULAÇÃO DE ARQUIVOS

Uma das coisas básicas em qualquer sistema operacional é o tratamento de arquivos. Neste capítulo serão vistos os principais comandos para a manipulação de arquivos: **ls, mkdir, rmdir, cd, pwd, cat, more, cp, mv, rm**.

Os nomes dos arquivos e diretórios podem ter tamanho de até 64 caracteres significativos. Um detalhe importante para se saber é o da existência de arquivos ocultos. Arquivos ocultos são os arquivos que não aparecem na listagem simples do diretório. O UNIX considera ocultos todos os arquivos que começam por um ponto.

Para um completo aproveitamento do que vai ser visto, são necessários alguns conceitos básicos:

- Entrada e Saída Padrão

Todo programa no UNIX tem uma entrada e uma saída. Geralmente a entrada padrão é o teclado (isto é, todos os dados que o programa precisa serão fornecidos via teclado) e a saída padrão é a tela do terminal (isto é, tudo que o programa gerar como resultado será mostrado na tela do terminal).

Mas pode-se mudar estes padrões utilizando o que se chama de redirecionamento. Os caracteres capazes de realizar isto são:

"<" modifica a entrada do programa

">" modifica a saída do programa

A forma de se utilizá-los segue abaixo:

programa > saída (tudo que for gerado será colocado em **saída**)

ou

programa < entrada (tudo que for necessário para o programa estará em **entrada**)

ou

programa < entrada > saída (o programa lê de **entrada** e escreve em **saída**)

Exemplos de como se usar o redirecionamento serão vistos no decorrer da apostila .

Metacaracteres

Para representar um conjunto de nomes de arquivos podem ser utilizados caracteres coringas (os metacaracteres), que são:

? representa um caracter

***** representa vários caracteres

Por exemplo:

a?e representa todos os arquivos que tem o nome de 3 caracteres sendo que o primeiro é **'a'** e o último é **'e'**. Todos os seguintes nome fariam parte do conjunto**: ale, ate, ave, ane e ame**. Mas, **alle** não.

com* representa todos os arquivos, independente do tamanho, que começam pela seqüência **'com'**. Esses arquivos fariam parte do conjunto**: com, comida, comando, compra.txt** e **compulsório**.

Este conceito ficará melhor entendido quando forem mostrados exemplos práticos.

4.1 LISTAGEM, CRIAÇÃO E MOVIMENTAÇÃO POR DIRETÓRIOS

Os arquivos no UNIX são organizados em diretórios. Cada diretório pode ter sub-diretórios formando uma árvore de diretórios. A árvore começa no diretório que é chamado de diretório raiz (representado por uma barra "/") e vai se expandindo tanto em profundidade como em largura.

A seguir serão vistos os comandos para visualizar o conteúdo de um diretório (**ls**), para a criação e remoção de diretórios (**mkdir** e **rmdir**), para a movimentação por entre os diretórios (**cd**) e para a visualização do nome e do caminho do diretório de trabalho atual (**pwd**).

Comando *ls*

Mostra o conteúdo de um diretório. Caso um arquivo seja especificado, **ls** mostra o seu nome e alguma outra informação requisitada. Quando nada é especificado o diretório atual é mostrado.

Sintaxe: **ls [-*opções*] [*nomes_de_arquivos_ou_diretórios*]**

onde as ***opções*** mais usuais são:

-a mostra todos os arquivos, inclusive os ocultos (começados por **.**)

-l mostra os arquivos com vários atributos (permissões, usuário, grupo, tamanho em bytes, data de última modificação, nome)

-R mostra recursivamente o conteúdo dos diretórios encontrados.

-F mostra como *ls* normal, mas indicado o tipo do arquivo:

- / : diretório
- * : executável
- @: link simbólico

Exemplos

Digite no prompt, **ls** e pressione a tecla <ENTER>, assim:

```
/home/pedro> ls
Materia   anual   classe   trabalho
/home/pedro> _
```

No exemplo acima foi mostrado o conteúdo do diretório corrente, que é o /home/pedro.

O mesmo exemplo, mas utilizando a opção **"-l"** ficaria assim:

```
/home/pedro> ls -l
drwxr-xr-x     daniel  grad93  24   Dec 13 16:47 Materia
-rw-------     daniel  grad93  2439 Dec 13 16:38 anual
-rw-rw-rw-     daniel  grad93  4226 Nov  7 18:53 classe
-rw-rw-r--     daniel  grad93  90   Dec 13 16:47 trabalho
/home/pedro> _
```

O primeiro caractere de cada linha indica o tipo do arquivo. Quando for **"d"**, indica que é um diretório, quando **"l"** indica que é um link e quando for **"-"** indica um arquivo. Os outros nove próximos caracteres indicam o modo de proteção do arquivo cuja explicação será vista no próximo capítulo. O próximo campo indica o proprietário do arquivo. Em seguida vem o nome do grupo ao qual pertence o proprietário, o tamanho em bytes, a data de última alteração e o nome do arquivo.

Com a opção **"-a"** conseguimos ver as entradas ocultas do diretório:

```
/home/pedro> ls -a
./    ../    .cshrc    .mailrc    .profile    Materia    anual    classe    trabalho
/home/pedro> _
```

Pode-se também utilizar combinações de opções, como:

```
/home/pedro>ls -la
drwxr-xr-x     3 daniel   grad93   1024      Dec 13 17:05 ./
drwx------     4 daniel   mail     1024      Dec 13 16:36 ../
-rw-rw-r--     1 daniel   grad93      6      Dec 13 17:04 .cshrc
-rw-rw-r--     1 daniel   grad93    123      Dec 13 17:05 .mailrc
-rw-rw-r--     1 daniel   grad93    123      Dec 13 17:05 .profile
drwxr-xr-x     2 daniel   grad93     24      Dec 13 16:47 Materia
-rw-------     1 daniel   grad93   2439      Dec 13 16:38 anual
-rw-rw-rw-     1 daniel   grad93   4226      Nov  7 18:53 classe
-rw-rw-r--     1 daniel   grad93     90      Dec 13 16:47 trabalho
```

onde foram listados os atributos de todos os arquivos e diretórios presentes no diretório atual.

E para visualizar o conteúdo do diretório **Materia**?

```
/home/pedro> ls Materia
notas       professor
/home/pedro> _
```

E para visualizar o diretório corrente e todos os sub-diretórios recursivamente?

```
/home/pedro> ls -R
Materia/    anual    classe    trabalho

./Materia:
notas    professor
/home/pedro> _
```

Neste caso primeiramente foi listado o conteúdo do diretório corrente e em seguida o do sub-diretório existente.

Agora usando os metacaracteres para ver todos os arquivos que contenham como terceira letra a vogal '**a**':

```
/home/pedro> ls ??a*
classe    trabalho
/home/pedro> _
```

Comando *mkdir*

Cria um novo diretório, cujo nome deve ser especificado.

Sintaxe: **mkdir [*nome_do_diretório*]**

Exemplo

Se é necessário a criação de um novo diretório no diretório corrente, deverá ser digitada a seguinte linha de comando:

```
/home/pedro> mkdir novo
/home/pedro> ls -F
Materia/    anual    classe    novo/    trabalho
/home/pedro> _
```

Comando *rmdir*

Remove o diretório especificado, que deve estar vazio.

Sintaxe: **rmdir [*nome_do_diretório*]**

Exemplo

Para remover o diretório novo do diretório atual deverá ser feito o seguinte:

```
/home/pedro> rmdir novo
/home/pedro> ls -F
Materia/    anual    classe    trabalho
/home/pedro> _
```

Comando *cd*

Muda o diretório de trabalho atual para o diretório especificado.

Sintaxe: **cd [*nome_do_diretório*]**

Exemplos

Os seguintes comandos mudam o atual diretório de trabalho para o diretório **Materia** e em seguida cria um novo diretório no mesmo:

```
/home/pedro> cd Materia
/home/pedro/Materia> mkdir Horarios
/home/pedro/Materia> ls -F
Horarios/    notas    professor
/home/pedro/Materia> _
```

Para ir ao diretório /home deve-se executar este comando:

```
/home/pedro/Materia> cd /home
/home> _
```

E agora para ir ao mais novo diretório criado (**Horarios**) fazer o seguinte:

```
/home> cd pedro/Materia/Horarios
/home/pedro/Materia/Horarios> _
```

Comando *pwd*

Mostra na tela o diretório de trabalho corrente.

Sintaxe: **pwd**

Exemplo

```
/home/pedro> pwd
/home/pedro
```

4.2 CRIAÇÃO E LISTAGEM DE ARQUIVOS

Os seguintes comandos têm por finalidade criar novos arquivos (uma das tarefas do **cat**), listar arquivos na tela ou em alguma outra saída especificada com o redirecionamento (**cat** e **more**) e concatenar arquivos (**cat**).

Comando *cat*

Realiza a concatenação de arquivos, ou cópia de arquivos ou simplesmente mostra o conteúdo dos arquivos na tela, dependendo dos argumentos fornecidos na linha de comando.

Sintaxes: **cat [*lista_de_arquivos*]** ou

cat [*lista_de_arquivos*] > [*arquivo*] ou

cat > [*arquivo*]

Exemplos

A primeira opção de sintaxe mostra o conteúdo de cada um dos arquivos na saída padrão (a tela). Por exemplo, suponhamos que existam os arquivos **arq1**, **arq2** e **arq3**. Para ver o conteúdo de **arq1** basta executar :

```
> cat arq1
este e o arq1
```

Caso se queira ver o conteúdo dos três arquivos juntos, digite:

```
> cat arq1 arq2 arq3
este e o arq1
este e o arq2
este e o arq3
```

A outra opção de sintaxe concatena os arquivos antes do sinal de redirecionamento e os coloca na saída especificada.

Caso queiramos criar o arquivo *arq123* que contenha o conteúdo dos três arquivos anteriores:

```
> cat arq1 arq2 arq3 > arq123
```

Para visualizar a concatenação basta utilizar o primeiro exemplo dado do **cat**:

```
> cat arq123
este e o arq1
este e o arq2
este e o arq3
```

A útilma opção de sintaxe coloca o que vier da entrada padrão (teclado) na saída especificada. Quando se acabar de digitar os dados que irão para a saída é necessário digitar a seqüência **“^d”** (CRTL-d) para indicar o término.

Por exemplo, criar um aquivo arq4 e colocar nele o seu nome e em seguida mostrá-lo na tela:

```
> cat > arq4
meu nome é pedro
<CTRL D>
> cat arq4
meu nome é pedro
```

Um outro sinal de redirecionamento existente é o >> que acrescenta os dados ao final da saída especificada, diferente do > que, primeiramente apaga o conteúdo da saída e depois coloca os dados.

Por exemplo, concatenar o **arq4** e o **arq1** e acrescentar ao **arq123** e depois mostrar o conteúdo de **arq123**:

```
> cat arq4 arq1 >> arq123
> cat arq123
este e o arq1
este e o arq2
este e o arq3
meu nome é pedro
este e o arq1
```

Comando *more*

Lista o conteúdo de um arquivo na tela, parando a cada vez que uma tela é preenchida. Ao final de cada tela aparecerá a mensagem **--More--**. Quando esta mensagem aparecer, pode-se apertar a barra de espaço para vizualizar a próxima tela de texto, ou pode-se teclar <ENTER> para vizualizar a próxima linha do texto ou pode-se digitar algum dos comandos que o **more** proporciona.

Sintaxe: **more [-*n*] [+*num_linha*][+/*expressão*] *arquivo(s)***

onde:

-n	usa uma janela de n linhas para mostrar o texto.
+num_linha	começa a mostrar o texto a partir da linha num_linha
+/expressão	começa a mostrar o texto duas linhas antes de aparecer expressão

Alguns dos vários comandos disponíveis são :

i<espaço>	mostra mais **i** linhas, se **i** não for fornecido, pula para a próxima tela.
CTRL D	mostra mais 11 linhas .
q ou Q	sai do **more**.
=	mostra o número da linha atual.
h	mostra todos os comandos disponíveis.
i/expressão	procura a i-ésima ocorrência de expressão.
in	procura a i-ésima ocorrência da última expressão procurada.
'	vai para a última posição onde se iniciou uma procura, se não foi realizada nenhuma, vai para o começo do texto.
!comando	executa o comando especificado.
:f	mostra o nome do arquivo atual e o número da linha atual.
.	repete o último comando.

Exemplo

Para vizualizar um texto chamado anual, com uma janela de 10 linhas e a partir de duas linhas acima da pimeira ocorrência de **"janeiro"** :

```
> more -10 +/janeiro anual
```

4.3 CÓPIA, REMOÇÃO E MOVIMENTAÇÃO DE ARQUIVOS

Os comandos que serão mostrados em seguida servem para tratar arquivos, como copiar um arquivo de um diretório para outro (**cp**), ou apagar algum determinado arquivo (**rm**) ou mover arquivos de um lugar para outro (**mv**).

Comando *cp*

Copia o conteúdo de um arquivo origem para um arquivo destino.

Sintaxe: **cp [-*ipr*] origem destino**

onde:

-i	pede confirmação para sobrescrever um arquivo já existente.
-p	copia não só o conteúdo do arquivo origem, mas tembém a data de modificação e as permissões.
-r	caso o arquivo origem seja um diretório e o destino também, copia o conteúdo de todo o diretório e sub-diretórios recursivamente para o diretório destino.

Exemplos

Para copiar o arquivo classe para o diretório **Materia**:

```
/home/pedro> cp classe ~/Materia
```

Agora fazer uma cópia de backup do arquivo **anual**:

```
/home/pedro> cp anual anual.bak
```

Fazer uma cópia de todos os arquivos do diretório **Materia** para o diretório **/tmp**:

```
/home/pedro> cp Materia/*  /tmp
```

Agora copiar todos os arquivos em que a quarta letra seja **u** para o diretório **/home/maria/tmp** :

```
/home/pedro> cp ???u* /home/maria/tmp
```

Comando *mv*

Move um arquivo de um lugar para outro ou renomeia arquivo, dependendo dos argumentos.

Sintaxe: **mv [-*fi*] nome novonome**

onde:

-f desconsidera qualquer restrição dos modo de proteção e também omite qualquer aviso a respeito desta violação.

-i pede confirmação para sobrescrever arquivo já existente.

Exemplos

Mover o arquivo **class** para o diretório **Materia**:

```
> mv class Materia
```

Agora renomear o arquivo **anual.bak** para **seguranca**:

```
> mv anual.bak seguranca
```

Agora renomear o nome do diretório **Materia** para **Materias**:

```
> mv Materia Materias
```

Comando *rm*

Apaga arquivo(s) de um diretório.

Sintaxe: **rm [-*fir*] arquivo**

onde:

-f não pede confirmação.

-i pede confirmação para cada arquivo.

-r remove o conteúdo de um diretório e seus sub-diretórios recursivamente

Exemplos

Remover o arquivo **seguranca** do diretório corrente:

```
> rm -i seguranca
seguranca: ? (y/n) y
```

Agora remover o arquivo **class** do diretório **Materias** inibindo as mensagens:

```
> rm -f  Materias/class
```

Agora remover todo o diretório **Materias** e seus possíveis sub-diretórios:

```
> rm -r Materias
directory Materias: ? (y/n) y
Materias/Horarios: ? (y/n) y
Materias/notas: ? (y/n) y
Materias/professor: ? (y/n) y
Materias: ? (y/n) y
```

5 SEGURANÇA

Todos arquivos/diretórios no UNIX tem uma proteção. Um arquivo pode ser configurado com relação à escrita e/ou leitura e/ou execução. As restrições podem ser feitas para o dono do arquivo, para o grupo (conjunto de usuários com características em comum) ao qual pertence o arquivo e para outros usuários que não sejam os dois anteriores. Para relembrar o que foi comentado no capítulo anterior, vamos mostrar alguns dos atributos de um arquivo:

-rwxrw-r-- 1 daniel grad93 90 Dec 13 16:47 carta.tex

Apenas foi citado que os nove caracteres (rwxrw-r--) seguidos do primeiro traço **'-'** (ou **'d'**) eram o *modo de proteção* do arquivo. E, este modo significa:

• Os três primeiros caracteres (**rwx**) representam as permissões do dono do arquivo (no caso, **daniel**).

r representa acesso para leitura (read) do arquivo.

w representa acesso para escrita (write) no arquivo.

x representa acesso para execução do arquivo.

• Os próximos três (**rw-**) representam as permissões do grupo.

- representa acesso negado para execução do arquivo.

• Os últimos três (**r--**) representam as permissões para os outros usuários, que no caso só têm permissão de leitura.

O próximo comando altera o modo de proteção de arquivos/diretórios.

Comando *chmod*

Muda o modo de proteção de um arquivo/diretório.

Sintaxe: **chmod *modo* arquivo**

O **modo** tem a seguinte forma:

[*quem*] *operador permissão* [*operador permissão*]

onde:

quem especifica-se para quem se está alterando a permissão, podendo ser o usuário (**u**), o grupo (**g**) ou outros (**o**).

operador pode ser **'+'** (para acrescentar a permissão) ou **'-'** (para retirar a permissão).

permissão pode ser para leitura (**r**), escrita (**w**) ou execução (**x**).

Exemplos

Supor que o diretório corrente tenha o seguinte conteúdo:

```
-rwx------     1 daniel   grad93      2439 Dec 13 16:38 anual
-rw-rw----     1 daniel   grad93      4226 Nov  7 18:53 classe
-rw-rw-r--     1 daniel   grad93       262 Dec 14 18:22 notas
drwxrwxrwx     2 daniel   grad93      1024 Dec 14 18:21 novidades
-rw-rw-r--     1 daniel   grad93       105 Dec 14 18:22 professor
-rw-rw-r--     1 daniel   grad93       367 Dec 14 18:22 sala
-rw-------     1 daniel   grad93      2439 Dec 14 18:58 seguranca
-rw-rw-r--     1 daniel   grad93        90 Dec 13 16:47 trabalho
```

Agora fazer as alterações :

• permitir acesso de leitura do **anual** para o grupo e para outros

```
>chmod og+r anual
```

• negar escrita do arquivo **classe** para o grupo

```
>chmod g-w classe
```

• negar acesso geral ao arquivo **trabalho** para grupo e outros

```
>chmod go-rwx trabalho
```

• permitir acesso geral ao arquivo **sala** para outros e grupo

```
>chmod go+rwx sala
```

Depois destas alterações, o diretório ficou assim:

```
-rwx---r--     1 daniel   grad93      2439 Dec 13 16:38 anual
-rw-r-----     1 daniel   grad93      4226 Nov  7 18:53 classe
-rw-rw-r--     1 daniel   grad93       262 Dec 14 18:22 notas
drwxrwxrwx     2 daniel   grad93      1024 Dec 14 18:21 novidades
-rw-rw-r--     1 daniel   grad93       105 Dec 14 18:22 professor
-rwxrwxrwx     1 daniel   grad93       367 Dec 14 18:22 sala
-rw-------     1 daniel   grad93      2439 Dec 14 18:58 seguranca
-rw-------     1 daniel   grad93        90 Dec 13 16:47 trabalho
```

6 OBTENDO AJUDA ON-LINE

Será abordado neste capítulo o comando **man**, que proporciona informações a respeito de uma diversidade de assuntos.

Comando *man*

No sistema UNIX há uma versão on-line do manual HP-UX Reference. O **man** mostra partes deste manual.

Sintaxe: **man *-k keyword*** ou

man *keyword*

No primeiro caso, pode-se informar aguma palavra-chave e ele retorna em que parte do manual ela está. No segundo, pode-se procurar por alguma parte inteira do manual, dando como parâmetro o nome do comando ou capítulo que se deseja ler.

Exemplos

Para ver qual parte do manual trata de **subtrees** :

```
> man -k subtrees
cp(1)          - copy files and directory subtrees
```

Agora para ver a parte do manual inteira a respeito do **cp**:

```
> man cp
```

O resultado desta operação fica como exercício. Examine também os comandos **ls, mv, pwd, cd**, etc.

7 COMANDOS DE PESQUISA

Comando *grep*

Este comando procura por uma expressão regular dentro de um ou mais arquivos.

Sintaxe: **grep [*opções*] *expressão* [*arquivos*]**

onde ***opções*** pode ser:

-c : Mostra o número de ocorrências da expressão.

-i : Ignora a diferença entre maiúsculas e minúsculas.

-n : Precede cada linha com seu número no arquivo.

-l : Lista apenas os nomes dos arquivos.

Exemplos

```
 > grep -ic amigo meu_texto
 > 4
 > _
```

Procura pela ocorrência de **'amigo'**, ignorando maiúsculas e minúsculas, no arquivo **meu_texto** informando o número de ocorrências.

```
> grep -il amigo *
meu_texto
carta
importante
> _
```

Procura pela ocorrência de **'amigo'**, ignorando maiúsculas e minúsculas, em todos os arquivos do diretório atual, informando o nome de todos os arquivos que possuem a expressão procurada.

Comando *find*

Este comando faz a procura de um determinado arquivo. É possível encontrar um arquivo pelo seu nome, nome do seu dono e pelo grupo ao qual pertença.

Sintaxe: **find *path [expressão]***

onde:

path: é o ponto de partida da procura;

expressão pode ser:

-name *nome_arq*: procura os arquivos com nome igual a *nome_arq*;

-user *nome_user*: procura os arquivos que pertencem ao usuário *nome_user*;

-group *nome_grupo*: procura pelos arquivos do grupo *nome_grupo*;

-print: mostra o path completo do arquivo

-ls: mostra todas as informações do arquivo, como tamanho, data, etc..

Exemplos

```
 >find /home/curso -name 'meu_arquivo' -print
 /home/curso/temp/meu_arquivo
 > _
```

Neste exemplo procuramos pelo arquivo **meu_arquivo** a partir do diretório /home/curso. Utilizando a opção **'-print'**, temos como resposta a localização completa do arquivo.

```
>find /home/curso -name 'ca*' -group 'pet' -ls
137384  1 -rw-r--r-- 1 julio pet   506 Jul 12 16:06 /home/curso/Mail/carta
92196   0 -rw-rw-r-- 1 julio pet   210 Dec 13 13:01 /home/curso/Pet/capitulo1
203108 12 -rw-rw---- 1 julio pet 23251 Jul 12 14:26 /home/curso/Pet/cap2.tex
>_
```

Neste segundo exemplo procuramos, a partir de /home/curso, todos os arquivos que começam com '**ca**' e que pertençam ao grupo **pet**. Utilizando a opção **'-ls'**, obtemos como resposta várias informações dos arquivos: permissões, dono, grupo a que pertence, tamanho, data de criação/atualização, etc.

8 COMANDOS GERAIS

Comando *quota*

Este comando indica o uso que o usuário tem do disco e seus limites.

Sintaxe: **quota [-v]**

O comando quota sem a opção **"-v"** exibe apenas um aviso indicando quota excedida; se esta for a situação do usuário. Caso sua quota exceda, você deve remover os arquivos dispensáveis das áreas indicadas, se isto não for possível, tentar compactar os arquivos; por exemplo, utilizando o comando *gzip*. É aconselhável verificar o diretório **/tmp** a procura de arquivos dispensáveis.

Exemplo

```
> quota -v
Filesystem  usage  quota  limit  timeleft  files  quota  limit  timeleft
/usr/soft   0      100    1000             63     0      0
/hp         13     100    1000             0      0      0
/hp/home    1901   4000   5000             202    0      0
>_
```

Comando *gzip*

Este comando compacta um ou mais arquivos. O uso de **'*'** e **'?'** é permitido.

Sintaxe: **gzip <arquivo(s)>**

Os arquivos especificados para compactação **não** serão armazenados em apenas um arquivo. Cada arquivo selecionado receberá a extensão **.gz**, indicando assim que se trata de um arquivo compactado.

Exemplo:

```
/home/pedro> ls
anual    classe      trabalho    Materia        carta
/home/pedro> gzip c*
/home/pedro>ls
anual    classe.gz   trabalho    Materia        carta.gz
/home/pedro>_
```

Comando *gunzip*

Utilize este comando para descompactar arquivos criados a partir do comando *gzip*.

Sintaxe: **gunzip <arquivo.gz>**

Exemplo

```
/home/pedro>ls
anual    classe.gz   trabalho    Materia        carta.gz
/home/pedro>gunzip classe.gz
/home/pedro>ls
anual    classe      trabalho    Materia        carta.gz
/home/pedro>_
```

Comando *pipe*

O comando '|' (**pipe**) tem como objetivo fazer a ligação da saída de um comando com a entrada de outro comando. Isto é chamado redirecionamento. A melhor forma de entender é com exemplo.

Exemplo

```
> ls -l
total 20
drwxr-xr-x 13     arpa   2560 Dec 12 17:21   arpa/
drwxr-xr-x 15  berutti   1024 Dec 12 18:27   berutti/
drwxr-xr-x  4    borba    512 Jul 27 1993    borba/
drwxr-xr-x  3     catj    512 Dec  7 21:27   catj/
drwxr-xr-x  3   clevan    512 Nov 11 20:47   clevan/
drwxrwxrwx  6    curso    512 Nov  8 19:33   curso94/
drwxr-xr-x  6    dehne   1024 Jan 27 1994    dehne/
lrwxrwxrwx  1     root     23 Sep 26 17:32   denilson/
drwxr-xr-x  8 dorothea    512 Jul 15 03:30   dorothea/
drwxr-xr-x 13     firk   1024 Dec 12 14:14   firk/
lrwxrwxrwx  1     root     21 Oct 18 10:01   foryta/
drwxr-xr-x  2    guest    512 Aug 25 19:17   guest/
lrwxrwxrwx  1     root     22 Jul 26 1993    irapuru
drwx--x--x  9   jcohen   1024 Oct 28 14:45   jcohen/
drwxr-xr-x  2    59000    512 Jun  3 1994    prppg/
drwxr-xr-x  5     rick    512 Sep 23 00:20   rick/
drwxr-xr-x  2   rnpnoc    512 Dec 17 1992    rnpnoc/
drwxr-xr-x  2      rpp    512 Nov 22 10:51   rpp/
> ls -l | grep 'Jul'
drwxr-xr-x  4    borba    512 Jul 27 1993    borba/
drwxr-xr-x  8 dorothea    512 Jul 15 03:30   dorothea/
```

```
lrwxrwxrwx  1     root     22 Jul 26 1993    irapuru/
```

Neste exemplo, o primeiro comando simplesmente listou todos os arquivos do diretório enviando o resultado para a tela. O segundo comando por sua vez, também listou os arquivos do diretório, mas enviando a saída para o comando **grep** (visto anteriormente) que selecionou, no nosso caso, aquelas linhas onde a expressão 'Jul' ocorria.

Comando *emacs* (Editor emacs)

Para utilizar o editor **emacs,** digite emacs na linha de comando. O editor mostrará um conjunto de informações bastante úteis. Como editor, possue as funções comuns e algumas bastantes avançadas. Para acessar estas funções devemos ter conhecimento de um conjunto de teclas. Abaixo está relacionada as mais usadas e importantes:

Obs.: O sinal de mais (+) não faz parte do comando. Representa associação de duas teclas, pressionadas juntas.

Ctrl+v	Rola uma página para baixo;
ESC+v	Rola uma página para cima;
Ctrl+l	Coloca o texto próximo ao cursor no meio da tela;
ESC+a	Vai para o início da sentença;
ESC+e	Vai para o fim da sentença;
ESC+<	Vai para o início do arquivo;
ESC+>	Vai para o fim do arquivo;
Ctrl+g	Desfaz um conjunto de teclas;
Ctrl+K	Apaga da posição do cursor até o final da linha (pode ser revertido);
Ctrl+y	Cola o que Ctrl+K cortou na posição atual do cursor;
Ctrl+x u	Desfaz uma ação anterior;
Ctrl+x Ctrl+f	Abre um arquivo;
Ctrl+x Ctrl+s	Salva o arquivo;
Ctrl+X Ctrl+c	Sai do emacs;
Ctrl+h	Exibe uma tela de help;
Ctrl+s	Procura para frente, proxima ocorrência Ctrl+s novamente;
Ctrl+r	Procura para trás, próxima ocorrência Ctrl+r novamente.

Comando *alias*

O comando **alias** tem como objetivo a definição de uma linha de comando.

Sintaxe: **alias [*nome [definição]*]**

onde:

nome: é o nome do *alias* que estamos criando;

definição: é uma lista de palavras que contém uma determinada função.

Se ***definição*** for omitida, o comando **alias** exibirá a definição corrente de nome. Se ***nome*** e ***definição*** forem omitidos, o comando **alias** exibirá todos os alias.

Exemplo

```
>alias dir ls -l
>alias dir
ls -l
drwxr-xr-x  4    borba    512 Jul 27 1993    borba
drwxr-xr-x  8 dorothea    512 Jul 15 03:30   dorothea
>_
```

Se setar um **alias** em linha de comando, como no exemplo, este só terá efeito nesta seção, ou seja, quando abandonar o sistema este **alias** será perdido. Porém, se quiser um **alias permanente**, o melhor lugar é em seu **.cshrc**.

9 PERSONALIZANDO ÁREA DE TRABALHO

Através da configuração das variáveis de ambiente no arquivo **.cshrc** configura-se a área de trabalho. Além de configurar as variáveis de ambiente utilizamos o arquivo **.cshrc** para inserir comandos que achamos necessários.

Algumas variáveis de ambiente:

PATH : Utilizada para configurar o caminho de procura dos programas executávesi;

MANPATH : Utilizada para indicar a localização dos arquivos de ajuda utilizados pelo man.

cwd : É o diretório corrente;

USER : Usuário corrente;

HOME : Define o diretório local (/home/curso);

SHELL : Define a shell padrão;

Um comando muito utilizado no arquivo **.cshrc** é o **alias**. Veja exemplo abaixo.

Fragmento de um arquivo **.cshrc** com algumas variáveis de ambiente configuradas:

```
....
set path = (/usr/local /usr/local/bin usr/bin /usr/etc /usr/bin/X11)
setenv MANPATH /usr/openwin/share/man:/usr/local/man:/usr/man
alias cd cd \!*;set prompt="`hostname` A:`echo $cwd`> "'
if ($?USER == 0 || $?prompt == 0) exit
.....
```

10 PROCESSOS

Como o UNIX é um sistema multitarefa, todo o seu funcionamento é através de processos. Um processo é um programa em execução, que utiliza-se de recursos da máquina em tempos determinados.

Comando *background* (&)

Um processo pode ser executado em primeiro ou segundo plano. Em primeiro plano é o que estamos mais acostumados. Quando digitamos um texto num editor estamos trabalhando em primeiro plano. Processos em segundo plano são normalmente processos que levarão muito tempo para serem executados, portanto, colocados em segundo plano para que não seja necessário ficar esperando até que ele termine sua execução e, possamos continuar usando o computador.

Para colocar um processo rodando em background basta que seja colocado um **'&'** no final do comando.

Exemplo

```
> who & > tmp
[1]3111
> cat tmp
dief    ttyp0 Dec 13 13:59 (harpia)
julio   ttyp1 Dec 13 14:11 (condor)
leandro ttyp2 Dec 13 14:23 (abutre)
jonatas ttyp3 Dec 13 14:32 (gaviao)
> _
```

Comando *ps*

O comando **ps** exibe o 'status' dos processos correntes.

Sintaxe: **ps [*opções*]**

onde **opções** pode ser:

-a : Inclue informações sobre processos pertencentes a outros.

-u: Exibe os campos: USER, %CPU, MEM, SZ, RSS e START.

-x: Inclue processos não associados com o terminal.

Exemplo

```
>ps -ef
USER       PID   %CPU    %MEM    SZ   RSS     TT   STAT    START    TIME    COMMAND
root       427   30.0    1.4     28   689     S    R       11:11    0:00    rlogin aguia
```

```
root      386   10.0    0.1    24   4      ?    S      15:07    0:00    in.rlogind
renato    372   0.0     0.0    25   60     pa TW       15:06    0:00    -bin/tcsh -i
renato    370   0.0     0.0    35   560    pa   TW     15:06    0:00    porlatex ec-t
julio     382 0.0       1.5    34   4460   p4 S        15:07    0:00    -tcsh (tcsh)
root      483 0.0       0.0    24   0      ?    IW     15:10    0:00    in.rlogind
root      472 0.0       0.1    48   24     p    6 S    15:11    0:00    rlogin aguia
>_
```

Sendo:

USER: Nome do dono do processo;

PID: Número inteiro identificador do processo;

%CPU: Quantidade de CPU utilizada pelo processo.

Comando *kill*

Este comando elimina um processo ativo pertencente ao usuário.

Quando eliminar processos:

- Quando um processo está provocando queda do desempenho;
- Um processo está 'pendurado';
- O terminal está travado.

Sintaxe: **kill PID**

onde **PID** é o número inteiro identificador do processo (veja comando **ps**)

Exemplo

```
>ps
3862 p4 S 0:01 -tcsh (tcsh)
5446 p4 S 0:00 sleep 20
5447 p4 R 0:00 ps
5042 p9 IW 0:00 -tcsh (tcsh)
>kill 5446
[1] Terminated sleep 20
>ps
3862 p4 S 0:01 -tcsh (tcsh)
5447 p4 R 0:00 ps
5042 p9 IW 0:00 -tcsh (tcsh)
>_
```

Com o comando **ps** identificamos o PID dos processos ativos. Utilizando o comando **kill**, eliminamos o processo 5446.

Este comando poderá ser útil também quando acontecer algum problema em seu programa que cause pane (trave). Para solucionar este problema basta logar-se em outra máquina, observar os processos que estão rodando na máquina (**ps)** que travou e eliminá-los (**kill**) .

11 COMPILAÇÃO DE PROGRAMAS EM 'C'

O sistema UNIX possui o compilador C como parte de seu sistema. O próprio UNIX foi desenvolvido em C. Para compilar programas em C basta usar o comando **cc** (em minúsculo).

Sintaxe: **cc [-o *arq_saida*] [-c] [-g] [-I *caminho*] [-L*Diretório*] *arq_origem***

onde:

-o : Gera o arquivo executável com nome *arq_saida*. Se a opção **'-o'** for suprimida será criado um arquivo executável **a.out**.

-c : Apenas compila gerando um arquivo com terminação **.o** para cada arquivo de entrada. Um único arquivo objeto pode ser gerado usando a opção '**-o'**.

-g : Para permitir a utilização posterior do depurador (" Debuger ");

-I : Adiciona *caminho* na lista de diretórios onde deve buscar os arquivos **#include**.

-L : Adiciona *Diretório* na lista de diretórios de bibliotecas.

arq_origem: é o nome(s) do(s) arquivo(s) que contém o programa fonte.

Exemplos

```
> cc -o teste teste.c
> ls
teste    teste.c
```

```
> _
```

Se nenhum erro for identificado o arquivo **teste** será criado e poderá ser executado em seguida, digitando **teste** na linha de comando.

```
> cc teste.c
> _
```

O arquivo **a.out** será criado pois não foi indicado um arquivo de saída, através da opção **'-o'**, neste segundo exemplo. O arquivo **a.out** também poderá ser executado a partir deste momento.

```
> ls
Mail/    Util/hello.c    a.out       magn    teste.c
Pet/     Xwindow/        carta.txt   mosaic
> a.out
Isto e' apenas um teste ...
>
```

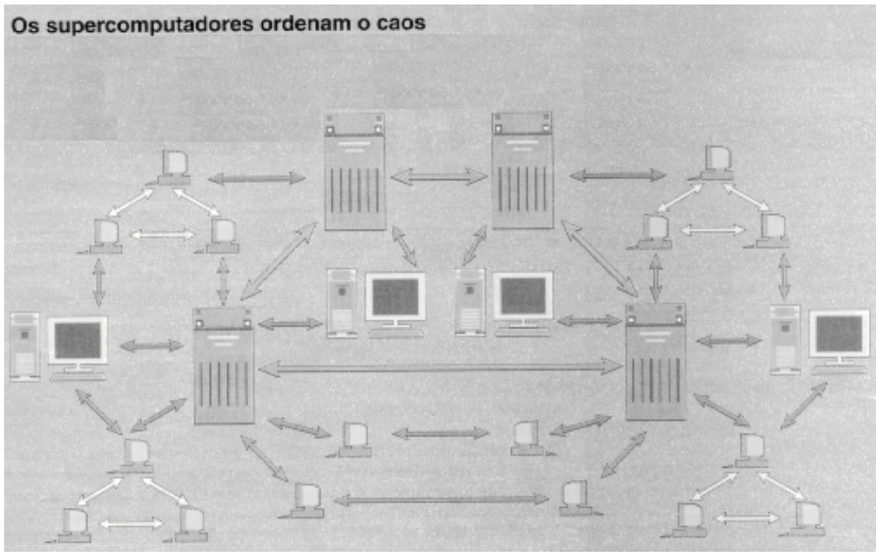
Parte II

INTERNET

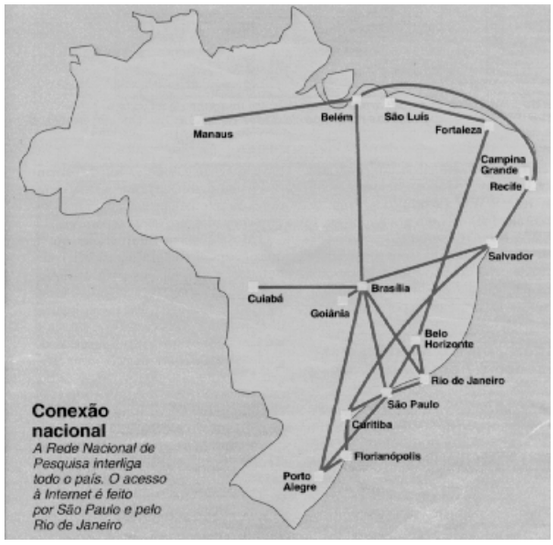
1 INTERNET

1.1 O QUE É INTERNET

A Internet, uma super-rede mundial de computadores, é uma cidade eletrônica onde há de tudo — museus, universidades, revistas, correio, bibliotecas, pesquisas. Ela já é freqüentada por mais de 30 milhões de pessoas, desde estudantes e pesquisadores, militares, governos, órgãos públicos, instituições educacionais, fãs de esporte ou de jazz, turistas, gente disposta a simplesmente conversar sobre qualquer coisa ou a discutir Física Quântica.



Parece uma metrópole. Mas não é. A Internet não existe fora dos fios e chips. Ela une computadores espalhados por todo o planeta. Na verdade, a Internet não é uma só: é a união de muitas redes existentes em vários países; podem se comunicar entre si porque utilizam o mesmo protocolo (TCP/IP) para troca de dados, ou seja, falam a mesma língua.



A necessidade da existência de redes distintas compondo a Internet decorre de vários aspectos. A tecnologia de redes limita a distância dos cabos. Diferentes centros usam diferentes tecnologias. Além disso, a segmentação reduz o tráfego de informação aumentando a eficiência e segurança.

A Internet começou a nascer na década de 60, no auge da Guerra Fria, quando o Departamento de Defesa americano imaginou uma maneira de proteger o sistema de comunicações em caso de ataque nuclear soviético, pois as estações de rádio, de televisão e telefônicas são os primeiros alvos de um borbardeiro. Em 1964, um pesquisador chamado Paul Baran projetou uma rede de computadores que não tinha uma central de controle de informações. A rede continuaria funcionando mesmo se algumas de suas partes fossem atingidas. As mensagens eram divididas em pacotes e enviadas em partes, para aumentar a segurança.

Nos anos 80, a NSF (Fundação Nacional de Ciência Americana) criou uma poderosa linha de transmissão, que se tornou a espinha dorsal da rede. Além dos militares, pesquisadores e grandes empresas também ganharam acesso à rede. Há pouco tempo, as portas da Internet foram finalmente abertas ao público. A Internet é muita ampla, cobrindo os Estados Unidos e extendendo-se pelo Canadá, Europa, Ásia e América do Sul. Estimativas do número de computadores servidores variavam de 40.000 a 500.000 e o número de usuários de 500.000 a um milhão, em 1988. Já em 1994, estes números saltavam para 2 milhões de computadores e mais de 30 milhões de usuários.

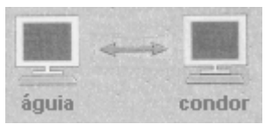
“O que tem na Internet? Não sei, a cada cinco minutos aparecem coisas novas”, diz Demi Getchko, gerente de informática da Fundação de Apoio à Pesquisa do Estado de São Paulo (Fapesp). A Fapesp é uma das duas únicas entidades pelas quais se faz acesso à Internet a partir do Brasil. A outra é a Universidade Federal do Rio de Janeiro. Essas linhas são pagas por órgãos

como o Conselho Nacional de Pesquisas (CNPq), que mantém a Rede Nacional de Pesquisa. Só para você ter uma idéia, a linha entre São Paulo e Chicago, nos Estados Unidos, custa 30.000 dólares ao mês. Por isso, só instituições acadêmicas têm acesso a essas linhas. O único serviço liberado a qualquer pessoa é o correio eletrônico, uma espécie de caixa postal usada para trocar mensagens.

"A Internet é a janela de um enorme e inusitado mundo novo, onde a diversidade encontra a singularidade. Um usuário inteligente saberá encontrar um universo de recursos informativos e um grupo infinito de interlocutores para suas preocupações, inquietações e elaborações intelectuais (...). Mas deverá estar atento para não sufragar diante tamanha avalanche de dados e possibillidades (...). A Internet é a maior revolução do século XX, constituindo-se numa infraestrutura indispensável tanto para a democratização da informação como a dissiminação viabilizante da participação democrática do cidadão no destino da sua civilização universal." — Prof. Ph. D. Gelson V. Gomes, UFPR.

1.2 CONCEITOS NECESSÁRIOS

Para efeito desta apostila consideraremos os terminais locais da Universidade Federal do Paraná **águia** e **condor**. E um terminal à distância em Chicago de nome **gargoyle.uchicago.edu**.



Qualquer processo de comunicação entre quaisquer 2 terminais envolve certas regras de endereçamento que tornam possível essa comunicação. Na Internet cada máquina tem um identificador numérico que é o seu real endereço Internet da rede. Para facilitar o uso define-se nomes para estes endereços numéricos.

Criou-se o conceito de domínios e subdomínios, formando uma denominação para a hierarquia da rede, onde:

Domínio	caracteriza a organização ou o país. Nos Estados Unidos, o nível mais externo caracteriza o tipo de organização **com** - empresa comercial **edu** - instituição educacional **gov** - organização governamental **mil** - organização militar **org** - organização não-governamental Exemplo: gargoyle.uchicago.**edu** é uma instituição educacional
	Em alguns outros países, o nível mais externo indica o código do país e o segundo, em geral caracteriza a organização. Abaixo códigos de países: **br** - Brasil **fr** - França **ca** - Canadá **au** - Austrália **uk** - Reino Unido

	de - Alemanha Exemplo: inf.ufpr.**br** é um subdomínio no Brasil
Subdomínios	indica uma sequência de um ou mais nomes de departamento Exemplo: **inf.ufpr**.br é um subdomínio do Depto de Informática na Universidade Federal do Paraná. Máquinas no domínio: **condor**.inf.ufpr.br **aguia**.inf.ufpr.br São máquinas no subdomínio **inf.ufpr**.br **gaviao**.inf.ufpr.br **fisica**.ufpr.br Depto de Física/UFPR **mat**.ufpr.br Depto de Matemática/UFPR **dcc**.ufmg.br Depto de Computação/UFMG

Também consideremos os usuários **usr1** e **usr2**, que terão acesso aos terminais locais (águia e condor). A identificação de um usuário é feita por meio de seu endereço eletrônico que, assim como no correio tradicional, apresenta uma estrutura própria. A forma de um enderreço eletrônico depende da rede a que pertence. Em geral é assim:

login@local.domínio

portanto, temos **usr1@inf.ufpr.br** e **usr2@inf.ufpr.br** como exemplos desta apostila.

Obs.: os alunos de Bacharelado em Informática têm o e-mail: ***login@aguia.inf.ufpr.br***

2 SERVIÇOS BÁSICOS

A Internet conta com um conjunto de serviços disponíveis para usuários. Nesta seção veremos alguns deles:

- Correio eletrônico
- Terminal remoto
- Transferência de arquivos
- Listas de discussão
- Informações de usuários da Rede
- Informações de conectividade da Rede
- Conversação entre dois usuários.

2.1 CORREIO ELETRÔNICO

O correio eletrônico constitui um dos mais importantes serviços de rede. É a aplicação que possibilita a comunicação dos usuários através da troca de mensagens.

O sistema de correio eletrônico, necessita de interfaces entre o usuário e o sistema, para facilitar o trabalho de todos, sobretudo do usuário. Essas interfaces são basicamente os programas ou utilitários de correio eletrônico, por exemplo o **elm** e o **mail** (trataremos aqui apenas do primeiro). Esses programas se encarregam de ajudar o usuário na composição/envio de mensagens. Ao mesmo tempo, têm a função de entregar e receber as mensagens do programa de gerenciamento do correio eletrônico.

Se você endereçar uma carta, escreverá o nome do destinatário, o nome de uma rua, o nº, a cidade, estado e CEP. Esta é toda a informação de que o correio necessita para enviar sua correspondência de modo razoavelmente rápido e eficaz. De modo análogo, os endereços de computador têm sua própria estrutura. A forma geral do endereço eletrônico (e-mail) de um usuário é: *login@local.domínio*. Portanto, o nosso usuário **usr1** tem seu e-mail **usr1@inf.ufpr.br**. Se for aluno de Bacharelado em Informática será **usr1@aguia.inf.ufpr.br**.

Caso deseje mandar uma mail para usuários que possuem conta na mesma máquina não será necessário digitar o e-mail completo, bastando utilizar somente o login do destinatário , sem o @ .

ELM

O programa **elm** implementa o sistema de correio eletrônico em máquinas UNIX, sendo de uso fácil e simples, pois fornece uma tela de interação com o usuário: as mensagens aparecem listadas e tem-se um menu de opções. Digite no prompt:

```
>elm
```

```
Mailbox is '/usr/spool/mail/usr1' with 2 messages [ELM 2.3 PL11]

 N  1  Aug 30 usr2@gargoyle.uchicago.edu  (133)      happy birthday
 N  2  Aug 24 tdk@ime.usp.br              (69)       reuniao


  You can use any of the following commands by pressing the first character;
d)elete or u)ndelete mail,  m)ail a message,  r)eply or f)orward mail,  q)uit
  To read a message, press <return>.  j = move down, k = move up, ? = help

Command:
```

Detalhemos as mensagens que chegaram:

```
N  1  Aug 30 usr2@gargoyle.uchicago.edu  (133)      happy birthday
```

A letra **N** e o nº **1** no canto esquerdo indicam que a mensagem **1** é nova e não foi lida ainda. Podemos encontrar outras letras, como:

D	mensagem deletada
E	indica uma mensagem vencida
N	identifica uma nova mensagem
O	mensagem já lida

Em seguida, vemos a data de emissão da mensagem (30 de agosto) e o remetente (usr2@gargoyle.uchicago.edu). O número que se encontra entre parênteses é apenas um identificador de pacotes. O texto `happy birthday` é o **subject** da mensagem, que é muito importante, pois indica brevemente de que se trata a mensagem.

Na parte inferior do **elm** pode-se observar o menu de comandos. Eis alguns comandos:

Comando	Função	Descrição
d	deletar; apagar	a mensagem é marcada e será apagada ao sair do programa
u	undeletar	recuperar uma mensagem marcada para ser deletada
s	salvar	esta opção requer o nome do arquivo, no qual a mensagem será salva
<enter>	ler	ler mensagem
m	enviar	requer endereço do destinatário. Para mais endereços, separar com vírgulas ou utilizar a opção **Copies to**. É requisitado o subject e, então o editor será aberto. Ao terminar de editar a mensagem, outro menu mosatrará as opções: editar (retorna ao editor), editar cabeçalho, enviar ou abandonar.
r	responder (reply)	responde diretamente ao remetente
f	repassar (forward)	usado para repassar a mensagem a outros usuários. Inclui toda a mensagem antiga, inclusive o cabeçalho.
q	sair	abandona o **elm**.

2.2 LISTA DE DISCUSSÃO

É um serviço especial do correio eletrônico que permite que uma correspondência seja distribuída à todos os assinantes do grupo. Este serviço permite a disseminação ágil de informações sobre a realização de eventos, chamadas de trabalhos para congressos, discussões técnicas e reuniões de trabalho. Tais grupos podem ser restritos ou públicos. Grupos restritos têm um coordenador que pode autorizar ou não a entrada de um assinante em potencial ao grupo. Grupos públicos têm acesso irrestrito, podendo qualquer usuário da rede participar deles por meio de mecanismos automáticos de inscrição.

As listas de discussão consistem na associação de um grupo de usuários, que possuem algum interesse em comum, com um nome que os agrupa e os identifica. Cada lista de discussão é um endereço eletrônico que redistribui para um conjunto de endereços, toda a correspondência recebida.

Há listas dedicadas a temas de qualquer área técnica e das ciências exatas, humanas e sociais. Existem máquinas servidoras (listserver) que administram várias listas. O cadastramento e a saída da lista são feitos através do envio de uma mensagem ao servidor da lista.
Para subscrever uma lista, é necessário conhecer duas informações: o **endereço do servidor da lista** e o **endereço da própria lista**.

O endereço de um servidor de listas costuma ser deste modo:

listserver@*subdomínios.domínio*

ou

mailserver@*subdomínio.domínio*

Para cadastrar-se em uma lista, basta enviar uma mensagem por correio eletrônico ao servidor da lista solicitando a sua inclusão na mesma, incluindo no corpo da mensagem:

subscribe *lista nome_real_usuário*

O usuário deve receber uma mensagem confirmando sua inclusão na lista.
Para ser retirado de uma lista, envie uma mensagem seguindo as mesmas regras de endereçamento, com a mensagem:

unsubscribe *lista*

Para enviar uma mensagem a lista, deve-se remetê-la para o endereço da própria lista.

mail *lista@subdomínios.domínio*

O usuário que enviou a mensagem poderá ter a confirmação de que a operação foi bem sucedida quando receber a cópia da mensagem.

2.3 TERMINAL REMOTO

Fornece acesso interativo a uma máquina remota a partir de uma máquina local, permitindo a utilização de todos os recursos da máquina remota. Este serviço permite utilizar as facilidades computacionais de outras instituições (supercomputadores, por exemplo) não disponíveis localmente. Os sistemas de terminal remoto a serem abordados aqui, são o **telnet** e o **rlogin**.

Telnet

Telnet é o protocolo do TCP/IP, usado na Internet para acesso a outros computadores ligados à rede, independentemente de sua localização física. Através do **telnet**, o usuário pode ter acesso a serviços disponíveis numa máquina remota, como se seu terminal estivesse conectado diretamente a ela. Para uso do serviço é necessário que o usuário possua uma conta no sistema remoto, embora vários deles possuam contas públicas, que podem ser acessadas por qualquer usuário da Internet.

A forma geral de uso do **telnet** é bastante simples. Basta digitar:

telnet *nome-do-computador-remoto*

Existem máquinas públicas que não exigem a identificação do usuário. Os sistemas de acesso público têm, em geral, interfaces auto-explicativas. No exemplo apresentado a seguir, uma sessão telnet é aberta na máquina gargoyle.uchicago.edu. Ao se estabelecer com o sistema remoto, é exibido o caracter de escape, no caso **^]**. Imediatamente, é exibida a tela de login, solicitando a identificação e a senha do usuário da máquina remota. Ao aparecer o prompt do sistema remoto, o usuário da máquina local passa a ter acesso a todos os recursos do usuário remoto. Digite:

```
> telnet gargoyle.uchicago.edu
```
abre uma conexão telnet

```
Condor: telnet gargoyle.uchicago.edu
Trying 139.82.17.20
Connect to gargoyle.uchicago.edu.
Escape caracter is '^]'.

SunOS UNIX (gargoyle)

login: usr1
Password:
Last login: Wed May 26 15:26:35 on console
Sun OS Release 4.1.1 (GENERIC) #1: Thu Oct 11 10:25:14 PDT 1990
You have new mail.
gargoyle:
```

nome do usuário
senha (não ecoada pelo sistema)

prompt da máquina remota

O programa telnet possui um modo especial de operação que oferece algumas funcionalidades a mais ao usuário. Pode-se entrar no modo comando de duas maneiras:
digitando simplesmente *telnet*, sem identificação da máquina remota ou
digitando o caracter de escape (CTRL-]) de dentro de uma sessão *telnet.*

Alguns comandos, os mais usados são:

close	encerra uma sessão corrente, reconectando o usuário da máquina remota. Causa o abandono da sessão aberta com telnet máquina. A maior utilidade deste subcomando reside no caso do usuário abrir sessões remotas sucessivas sem perder o caminho;
open máquina-remota	abre uma conexão na máquina especificada;
quit	abandona o programa telnet;
status	informações sobre o estado da conexão
<ENTER>	(vazio) coloca o usuário no modo de comandos na máquina remota.
?	mostra informações de help

Exemplo:

```
> telnet                                          entra no telnet em modo comando
```

```
telnet> ?                                         lista comando disponíveis
Commands may be abbreviated. Commands are:
close          close current connection
...
z              suspend telnet
?              print help information
telnet> open sonne.uiuc.edu                       abre uma conexão telnet
Trying ...
Connected to sonne.uiuc.edu
..
..
sonne> quit                                       sai do telnet e retorna ao sistema local
>
```

Rlogin

O programa **rlogin** é utilizado para servidores em sistemas UNIX, onde o parâmetro usuário é a identificação da máquina remota, sendo desnecessário para se conectar na mesma conta em máquinas de uma mesma instalação. Sua sintaxe:

rlogin *[-l usuário] máquina.subdomínios.domínio*

No exemplo a seguir, o usuário logado na máquina local (águia) solicita uma sessão com outra máquina local, onde também tem conta. Por esse motivo, essa informação não é solicitada pelo sistema.

```
HP-UX aguia

login: usr1                                       o usr1 entra no sistema na máquina águia
Password:                                         senha (não ecoada pelo sistema)
Last login: Fri Dec 16 14:07:42 from galileo


aguia> rlogin condor                              faz conexão remota com a condor
Last login: Fri Dec 17 09:15:53 from aguia
SunOS Release 4.1.2 (GENERIC_SLIP_POSTGRES) #1: Thu

condor>_                                          agora, pode-se fazer uso dos recursos alocados a condor
```

Aqui, é estabelecido uma sessão remota em Chicago. No caso, é fornecida no próprio comando, a identificação do usuário remoto, e é solicitado a senha.

```
HP-UX aguia

login: usr1
Password:
Last login: Fri Dec 16 14:07:42 from galileo

aguia> rlogin gargoyle.uchicago.edu -l usr1
Password:
Last login: Fri Dec 21 13:15:03 from aguia
SunOS Release 4.1.2 (GENERIC_SLIP_POSTGRES) #1: Thu

gargoyle>_
```

2.4 TRANSFERÊNCIA DE ARQUIVOS

Permite transferir programas, dados, textos, figuras, etc. entre os equipamentos da rede (sendo bastante útil para a realização de trabalhos cooperativos). O próprio usuário interessado comanda, a partir de seu computador, a transferência dos arquivos remotos (para os quais foi dada permissão de leitura) que desejar.

FTP

O protocolo **ftp** (File Transfer Protocol) é o principal método de transferência de arquivos na Internet. Através desse serviço, e com a devida permissão, é possível copiar um arquivo de ou para qualquer máquina ligada à Internet. Vários sistemas oferecem serviço de ftp anônimo, que permite que qualquer pessoa tenha acesso a arquivos públicos com facilidade. Alguns sistemas têm discos ou mesmo computadores inteiros dedicados a manter enormes *repositórios* com programas e bases de dados com informações diversas, como é o caso do *gatekeeper.dec.com* (Digital), *wuarchive.wustl.edu* (Washington University in Sant Louis), *archive.cis.ohio-state.edu* (The Ohio State University) e *simtel20.army.mil*.
A forma geral do comando é:

ftp *nome-do-sistema-remoto*

No caso de ftp anônimo a pessoa entra no sistema remoto como usuário **anonymous**. É costume que se responda à solicitação de senha (Password:) com o endereço eletrônico do usuário, embora isso não seja obrigatório. Uma vez dentro do ftp, vários comandos estão disponíveis, veja alguns:

Função	Comandos	Exemplos
Listagem de Diretórios	**ls** Fornece uma listagem simples do diretório	ftp> ls 200 PORT command successful 150 Opening ASCII mode data connection for file list nefnet scott campus
	dir Produz uma listagem completa do diretório	ftp> dir 200 PORT command successful 150 Opening ASCII mode data connection for /bin/ls total 3116 -rw------- 1 krol cso 110 Oct 31 08:18 nefnet -rw-r--r-- 1 krol cso 21 Nov 21 15:11 scott drwx----- 2 krol cso 120 Mar 5 17:33 campus
Mudando de diretório	**pwd** Permite saber qual o diretório corrente da máquina remota	ftp> pwd /mnt/pub
	cd Muda de diretório na máquina remota	ftp> cd ietf 250 CWD command successful
	lcd Muda de diretório na máquina local	
Transferindo arquivos *	**Modos de transferência** **ascii** (default) A tranferência é tratada como um conjunto de caracteres. Arquivos de texto são sempre transferidos em modo ascii.	ftp> ascii 200 Type set to A
	binary A sequência de bits do arquivo é preservada de forma que a cópia e o original são idênticos, bit a bit. Arquivos executáveis são transferidos neste modo.	ftp> binary 200 Type set to I
	Comandos de Transferência **get *nome_do_arquivo_remoto*** Transfere um arquivo do computador remoto para o sistema	ftp> ls b* 200 PORT command successful 150 ASCII data connect for /bin/ls

	local **put *nome_do_arquivo_local*** Transfere um arquivo do sistema local para o computador remoto. **mget *lista_de_arquivos*** Transfere grupo de arquivos do sistema remoto para o local **mput *lista_de_arquivos*** Transfere grupo de arquivos do sistema local para o remoto	b.tst bash.help bsdman.sh 226 ASCII transfer complete remote:b* 29 bytes received in 0.03 seconds ftp> mget b* mget b.tst? yes 200 PORT command successful 150 ASCII data connect for b.tst 226 ASCII transfer complete local: b.tst remote:b.tst 81927bytes received in 0.41 seconds mget bash.help? no mget bsdman.sh? no ftp>
Comandos locais	Quase todos os comandos remotos (ls, dir, cd, pwd) têm a sua versão local, bastando-se adicionar "!" no começo da expressão	ftp> !pwd *ou* ftp> !cd
Comandos diversos	close Fecha uma conexão ftp e retorna modo comando quit termina uma sessão de ftp e retorna ao sistema operacional help Informaçòes resumidas dos comandos de ftp open endereço abre uma conexão ftp com a máquina especificada user usuário define novo usuário sobre a mesma conexão	ftp> close ftp> quit ftp> help ftp> open gargoyle.uchicago.edu ftp> user user2

* Antes de efetuar uma transferência, certifique-se do tipo de arquivo a ser copiado. A tabela abaixo dá algumas dicas:

FILE	MODE
Arquivos texto	ascii
Código fonte de programas	ascii
Arquivos executáveis	binary
Arquivos comprimidos	binary
Arquivos postscript	ascii
Mensagens eletrônicas	ascii
Arquivos Unix "tar"	binary
Arquivos Unix "script"	ascii
Arquivos backup	binary

Transferência anônima

Existe uma forma de transferência especial usada para recuperar cópias de software, artigos, e material de acesso público em geral. Ela é chamada de "transferência anônima".
Nesse caso, usa-se uma máquina pública, e fornece-se "anonymous" como identificação, e o endereço eletrônico do usuário que está solicitando o serviço, como senha (por etiqueta).
O procedimento, uma vez dentro da sessão usando a identificação da máquina remota é o mesmo da transferência comum.

A diferença básica está no escopo de ação. Nesse tipo de transferência, o usuário passa a ter acesso a um diretório que acima do qual não consegue navegar. No ftp anônimo não é permitido, também, realizar qualquer operação que implique modificações na estrutura de remota de arquivos. Exemplos:

```
> ftp ftp.uu.net
```

```
Connect to ftp.uu.net
220 uunet FTPserver (Version 5.100 Mon Feb 11 17:13:28 EST 1994) ready.
Name (ftp.uu.net:jm) : anonymous
331 Guest login ok, send ident as password
Password: usr1@inf.ufpr.br
230 Guest login ok, acces restrictions apply.
```

```
> ftp wuarchive.wustl.edu
```

```
Connect to wuarchive.wustl.edu
220 wuarchive.wustl.edu FTPserver (Version 5.100 Mon Feb 11 17:23:28
EST 1994) ready.
Name (wuarchive.wustl.edu:peter) : anonymous
331 Guest login ok, send your complete e-mail address as password
Password: usr1@inf.ufpr.br
230 Guest login ok, acces restrictions apply.\

ftp> ls
PORT command successful
150 Opening ASCII mode data connection for file list.
etc
pub
doc
README
226 Transfer complete.
176 bytes received in 0.015 seconds (11 Kbytes/s)

ftp> get README
200 PORT command succesful.
150 Opening ASCII mode data connection for README (2928 bytes).
226 Transfer complete.
local: README remote: README
2988 bytes received in 24 seconds (0.12 Kbytes/s)

ftp> quit
221 Goodbye.
```

2.5 INFORMAÇÕES DE USUÁRIOS E CONECTIVIDADE DA REDE

Há serviços que possibilitam encontrar localização e identificação de objetos da rede, sendo estes, os de maior interesse para os usuários, máquinas e e-mail de outros usuários.

Finger

O comando **finger** fornece informações sobre usuários de um sistema e Sua sintaxe é:

finger usuário@máquina

onde *usuário* é opcional e especifica o nome a ser pesquisado. Por exemplo, para obter informações sobre o usuário **usr2** na máquina **gargoyle.uchicago.edu**, digite:

```
> finger usr2@gargoyle.uchicago.edu
```

```
[gargoyle.uchicago.edu]
Login name: usr2                            In real life: Exemplo da Internet II
Directory:/home/visit/usr2           Shell:/bin/csh
Last login Mon Dec 3, 1990 on ttyq5 from dc-mac49
No plan
>_
```

Finger é frequentemente usado para obter uma lista das pessoas que estão correntemente usando um sistema remoto. Para isso, basta omitir a parte *usuário* do comando.
Assim:

```
> finger @clotho.fapesp.br
```

```
[clotho.fapesp.br]
Login          Name                 Tty     Idle    When           Where
hugo           Hugo Pena            p0      1:39    Wed 08:49
wilson         Wilson Sarto         p2      1:05    Wed            lakesis
zegonc         Jose Goncalves       p5      .....
```

Finger sem argumentos exibe um sumário de quem está conectado no sistema local. Por razões de segurança, algumas instalações obstruem, para o mundo externo, o acesso às informações sobre os seus sistemas e usuários através do *finger*.

Ping

Permite ao usuário verificar se um sistema está em funcionamento em um determinado instante. O exemplo abaixo verifica se a máquina está ou não funcionando:

```
> ping gargoyle.uchicago.edu
gargoyle.uchicago.edu is alive
> _
```

ou

```
> ping aguia.inf.ufpr.br
no answer from aguia.inf.ufpr.br
> _
```

2.6 CONVERSAÇÃO ENTRE DOIS USUÁRIOS

Dois usuários conectados à Internet pode conversar de forma eletrônica, através do vídeo e do teclado. O usuário solicita a comunicação remota com o usuário. Este recebe um aviso no seu vídeo. Se quiser responder ao chamado, será estabelecido um caminho de comunicação através de dois canais entre os dois usuários, que passam a digitar e receber as intervenções no teclado e no vídeo, respectivamente. Caso o usuário remoto, não queira responder, bastará ignorar a chamada.

Talk

O comando *talk* permite que dois usuários mantenham uma conversação interativa, em tempo de resposta instantânea, ao invés do estilo escreve-espera do correio eletrônico. Por exemplo, se o usuário **usr1** quiser falar com o usuário **usr2** na máquina **gargoyle.uchicago.edu** ele procederá da seguinte maneira:

```
> talk usr2@gargoyle.uchicago.edu
```

Se **usr2** estiver usando o sistema, aparecerá no seu terminal uma mensagem assim:

```
Message from Talk_Daemon@gargoyle.uchicago.edu at 21:25...
talk: connection requested by usr1@aguia.inf.ufpr.pr
talk: respond with: talk usr1@aguia.inf.ufpr.pr
```

Se **usr2** aceitar e responder ao pedido de conexão, o terminal de ambos os usuários ficará assim:

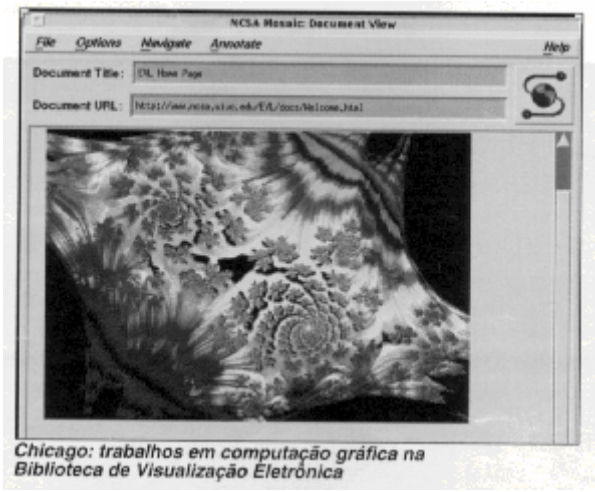
```
[Connection established]


------------------------------------------------------------
```

E a conversação pode ser iniciada, digitando-se as mensagens normalmente, pois a comunicação é “full-duplex”, ou seja, simultânea como num telefone. Pode-se receber caracteres ao mesmo tempo que envia-se outros. Para terminar, digite **CTRL-C**.

3 APLICAÇÕES AVANÇADAS

3.1. WWW



Chicago: trabalhos em computação gráfica na Biblioteca de Visualização Eletrônica

A Internet, no começo, não oferecia tantas facilidades aos principiantes. Tim Berners-Lee, pesquisador do CERN (Centro Europeu de Pesquisa Nuclear), cansou-se de ter que acessar um computador diferente cada vez que usava um dos serviços da Internet. Em 1989, ele criou um programa que integrava todos os sistemas, e o chamou de WWW (World Wide Web). O WWW não só integrou tudo, como permitiu usar multimídia e hipertexto.

Mas, era preciso criar uma interface para esse programa. Assim, qualquer pessoa poderia utilizá-lo. No início de 1993, o NCSA jogou na rede o Mosaic, a interface que transformava as antigas telas, cheias de comandos complicados, em agradáveis páginas com cores, fotos e comandos facílimos de usar. O sucesso foi estrondoso. Em apenas um ano, o número de pessoas que acessaram Mosaic cresceu 220 000%.

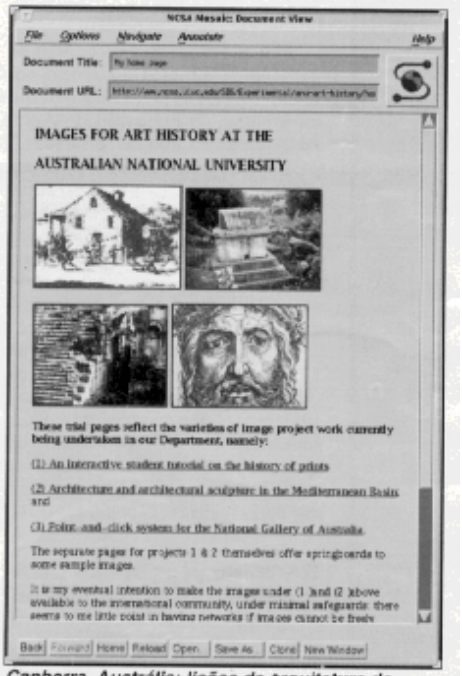
Com o Mosaic, uma pessoa que entra no banco de dados da NASA não lê apenas um texto, mas pode ver imagens de satélite ou fotografias da Terra tiradas pela tripulações dos ônibus espaciais. Pode fazer turismo em Paris e entrar no Museu do Louvre. Pode ainda ler revistas. Basta digitar:

```
> mosaic
```

Para acessar outros documentos, é preciso saber os caminhos onde estão armazenados tais documentos. De posse deles, abra a opção *Open URL* do menu *File* e digite-os.

Alguns Internet World Wide Web Sites:

Begginer's Guide to HTML	http://www.ncsa.uiuc.edu/demoweb/html-primer.html
Honolulu Home Page	http://www.hcc.hawaii.edu
National Center for Atmospheric	http://http.ucar.edu/metapage.html
SSC Home Page	http://www.ssc.gov.SSC.html
Vatican Exhibit	htttp://www.ncsa.uiuc.edu/SDG/Experimental/vatican.exhibit/Vatican.exhibit.html
FTP Sites	http://hoohoo.ncsa.uiuc.edu:80/ftp-interface.html
British Columbia	http://www.cs.ubc.ca/
Virtual Tourist - Califórnia	http://www.research.digital.com/SRC/virtual-tourist/California.html
Internet talk radio	http://www.ncsa.uiuc.edu/radio/radio.html



Canberra, Austrália: lições de arquitetura do Mediterrâneo e uma bela galeria de arte

3.2 GOPHER

O programa Gopher é um software de recuperação de documento que fornece uma interface dirigida por menu que auxilia o usuário na busca de informações. O Gopher é muito útil para usuários sem experiência que não sabem exatamente o que procurar mas necessitam de alguma informação da rede.

Além de ajudar na pesquisa, o Gopher pode ainda transferir esta informação para o usuário interessado. Para executá-lo digite:

```
> gopher
```

```
Internet Gopher Information Client v0.8

              Root Directory
      -->   1. Welcome to U the Illinois Gopher
            2. CCSO Documentation/
            3. Computer References Manual/
            4. Frequently Asked  Questions/
            5. GUIDE to U of Illinois
            6. Libraries/
            7. National Weather Service/
            8. Other Gopher and Information Servers/
            9. Peruse FTP Sites/
            10. Phone Books/

Press ? for Help, que to Quit, u to go up
```

O menu é auto-explicativo e você pode utilizar as setas para cima e para baixo para mover o apontador (-->) e pressionar RETURN para selecionar a opção, ou então, digitar o número da opção desejada.

3.3 WAIS

O servidor Wais (Wide Area Information Server) é um recuperador de informações similar ao Gopher, que ajuda o usuário a achar e recuperar documentos e outras informações. A diferença é que o Wais se destina a usuários mais avançados que sabem o que querem, embora possam não saber onde localizar. O Wais tem a capacidade de localizar um documento baseando-se no conteúdo do mesmo, fazendo busca a partir de palavras-chave ou qualquer outra cadeia de palavras contidas em um documento.

Para usá-lo, faz-se um *telnet* para uma máquina que o possua e depois responde-se ao login com a palavra *wais*, deste modo:

```
> telnet quake.think.com
```

```
Trying 192.31.181.1...
Connected quake.think.com.
Escape character is '^]'

SunOS UNIX (quake)

login: wais
Welcome to wais
Please type user identifier: usr1@inf.ufpr.br
Term = vt100
Starting wais ...
```