# FMA 215

## Aula 14: Arquivos Tar e Backups

Prof. Jorge L. deLyra

26 de Agosto de 2002

## 1 Alguns Conceitos Relevantes

O comando `tar` é um dos mais úteis e versáteis do sistema. Ele é usado para criar, ler e manipular arquivos tar, que são arquivos que contém outros arquivos ou mesmo diretórios completos. Em inglês chama-se a um arquivo tar de "archive", enquanto os arquivos comuns são chamados de "files", mas em português o mesmo nome é usado para as duas coisas. Um arquivo tar é tanto um "file" quanto um "archive". O primeiro nome refere-se ao fato de que o arquivo tar é um arquivo dentro de um filesystem, exatamente como qualquer outro. O segundo nome descreve o seu conteúdo, que é o de um arquivo de arquivos. Nesta apostila vamos traduzir "file" por "arquivo" e "archive" por "arquivo tar".

O comando tar pode ser usado para fazer backups em fita, de fato o seu nome vem de "tape archiver", ou seja, arquivador em fitas. Mas ele também pode ser usado para empacotar arquivos ou mesmo diretórios inteiros em um único arquivo tar em disco, para copiar árvores de diretórios de uma lugar para o outro, para fazer backups em disquetes e ZIPs e replicar filesystems inteiros. Sua ação sobre diretórios que sejam dados como argumentos é recursiva, ou seja, o comando `tar` percorre recursivamente os subdiretórios do diretório dado, indo até o final da árvore de diretórios que se inicia nele. Em suma, é um comando de extraordinária flexibilidade e utilidade para a manipulação de arquivos.

Esta ação recursiva sobre diretório, que constitui uma forma de empacotar, desempacotar e replicar quantidades quase ilimitadas de dados, inclusive filesystems inteiros, nos permite realizar facilmente a clonagem de sistemas operacionais completos. Esta é uma forma muito rápida e fácil de se instalar um novo sistema de forma idêntica a um que já exista, poupando enormes quantidades de trabalho de instalação e, principalmente, de configuração de sistemas que sejam muito parecidos. Será realizada uma demonstração deste tipo de operação durante a aula, usando um método um pouco diferente daquele que veremos explicitamente nesta apostila, um método que envolve o conceito de sub-shell.

O programa `tar` é uma das ferramentas mais úteis para a manutenção e gerenciamento de sistemas. Além da possibilidade de usá-lo para fazer backups dos filesystems de sistema, que são basicamente estáticos, é possível usá-lo em combinação com outros elementos e aspectos do sistema para fazer, por exemplo, backups automáticos em disco das contas dos usuários, que em geral são muito dinâmicas. Pode-se fazer uma atualização automática do backup, que denominamos de "backup incremental", digamos de hora em hora, de forma que o usuário tenha certeza de que nunca vai perder qualquer arquivo que tenha mais de uma hora de idade. Nos sistemas do projeto Sócrates temos hoje um sistema de backup incremental automático das contas dos usuários, que roda de quatro em quatro horas. Além disso, é possível fazer backups de segurança através da rede de áreas vitais do sistema, tais como os logs e as configurações, o que é muito útil como uma forma de controlar invasões hostis dos sistemas.

Nesta aula vamos ilustrar alguns destes usos do comando `tar`, incluindo arquivos tar em disco e em disquetes. A única coisa que *não* vamos fazer é escrever em fitas, pois esta é uma atividade que está ficando progressivamente mais rara para um usuário comum. Com a queda dos preços dos discos rígidos, hoje em dia é mais fácil, conveniente e barato fazer backups em disco do que em fitas. Para o transporte de dados, os ZIPs são uma das melhores escolhas, mas também é possível usar floppies comuns, escrevendo neles arquivos tar em modo multi-volume, ou seja, um único tar em uma coleção de disquetes, como veremos.

Com o aparecimento e a popularização de unidades gravadoras de CDROM e de CDRW, tornou-se disponível uma nova forma de armazenamento de dados, muito barata e bastante conveniente, tanto para backups quanto para o transporte de dados. Já temos uma unidade deste tipo disponível na sala Pró-Aluno e em breve teremos mais duas. Isto nos permite escrever discos de cerca de 650 MB com mídias que custam entre cerca de R$ 1 (para as que se escreve uma única vez) e cerca de R$ 5 (para as que podem ser re-escritas muitas vezes). Já estão começando a aparecer no mercado unidades de gravação de DVD, de forma que em breve poderemos estar escrevendo discos deste tipo com mais de 4 GB cada um.

Por ora ainda é necessário efetuar operações relativamente complexas com comandos especiais para gravar estes discos óticos, mas já existem módulos experimentais para o kernel do Linux que nos permitem tratá-los exatamente como tratamos hoje discos ZIP ou floppies. Com a melhoria progressiva das velocidades de gravação e o aumento de capacidade das mídias, já podemos antever uma época, num futuro próximo, onde será possível manipularmos e armazenarmos quantidades praticamente ilimitadas de informação em discos óticos baratos e confiáveis, de forma muito simples e transparente.

# 2 Tarefas

1. Vamos começar criando um arquivo tar simples, contendo um subdiretório de sua conta. Presumivelmente você tem em sua conta um diretório chamado `WWW` com pelo menos alguns arquivos dentro. Se este não for o caso, use algum outro diretório. Estando na raiz de sua conta, você pode criar um arquivo tar contendo este diretório com

   ```
   tar -cf my-web-hp.tar WWW
   ```

   A opção `-c` instrui o comando a criar um novo arquivo tar. Observe que `my-web-hp.tar` é o argumento da opção `-f`, que define qual é o arquivo a ser criado. O nome do arquivo é arbitrário, enquanto a terminação `.tar` é um padrão que indica tratar-se de um arquivo tar. Observe que, no caso de uma opção como esta, que requer um argumento, o argumento deve sempre seguir imediatamente a opção na linha de comando. Verifique, usando o comando `ls`, que o arquivo tar foi de fato criado. O argumento do comando `tar` é o diretório `WWW`, que define qual vai ser o conteúdo do arquivo tar.

   Note que, curiosamente, algumas opções do comando `tar` não são de fato completamente opcionais. É necessário que seja usada uma opção que determine que tipo de operação o comando vai realizar: escrita, leitura, tabela de conteúdos, etc. É o caso aqui da opção `c`, assim como das opções `x`, `t`, `r` e `u`, como veremos mais tarde. É necessário que uma e apenas uma destas opções alternativas esteja presente a cada vez que se usa o comando.

2. Para se garantir contra erros, seria uma boa idéia criar um segundo arquivo deste tipo, contendo o diretório `WWW` e que será mantido como backup caso algo aconteça com ele. Sem se preocupar com os detalhes, que serão explicados ao longo das tarefas, execute o comando

   ```
   tar -czpf WWW.tgz WWW
   ```

   Em seguida, para garantir que este arquivo de backup não seja apagado ou sobrescrito acidentalmente, vamos protegê-lo contra escrita usando o comando

   ```
   chmod ugo-w WWW.tgz
   ```

   Com isto, você poderá ler o arquivo para recuperar seus dados em caso de necessidade, mas não poderá apagá-lo ou escrever algo em cima dele.

3. Observe que o comando `tar` funcionou silenciosamente na tarefa anterior, sem dar qualquer mensagem exceto erros. Para tornar a sua ação mais visível podemos usar a opção `-v` (de verbose), que instrui o comando a mostrar ao longo do "run" cada arquivo que é acrescentado ao tar. Repita o comando da forma

   ```
   tar -cvf my-web-hp.tar WWW
   ```

   Observe que este comando sobrescreveu o arquivo já criado anteriormente sem dar qualquer aviso. Naturalmente, isto não é grave neste caso, pois acabamos de sobrescreve-lo com exatamente o mesmo conteúdo, mas fica um aviso de cuidado ao se criar e escrever arquivos com o comando `tar`.

4. Mesmo sem ter usado a opção `-v` para escrever o arquivo podemos descobrir qual é o seu conteúdo usando a opção `-t`, que extrai dele a tabela de conteúdos. Faça isto usando o comando

   ```
   tar -tf my-web-hp.tar
   ```

   É possível obter informações mais detalhadas sobre os arquivos contidos no arquivo tar combinando isto com a opção `-v`, que dá um resultado no estilo de `ls -l`:

   ```
   tar -tvf my-web-hp.tar
   ```

   Desta forma podemos saber que tipo de arquivos existem no tar, quais os seu nomes, a quem pertencem e qual o seu tamanho, todas elas informações muito úteis para se poder retirar dele o que se quiser.

5. Observe que todos os arquivos estão guardados dentro do tar dentro de um diretório `WWW`. Isto significa que os arquivos estão guardados com paths relativos, começando por este diretório. Quando tentarmos extrair os arquivos, este diretório será criado, caso ainda não exista, como um subdiretório do diretório onde estivermos. Esta é uma boa forma de registrar, usando o nome deste diretório, qual é a natureza geral do conteúdo do arquivo tar. Mas podemos evitar colocar este diretório no tar, caso isto não seja conveniente por qualquer motivo. Para fazer isto, entre no diretório `WWW` e sobrescreva o arquivo mais uma vez usando

   ```
   tar -cvf ../my-web-hp.tar .
   ```

   Desta vez estamos informando ao comando que o arquivo tar deve ser criado no diretório de cima e que o conteúdo será o diretório corrente, representado pelo ponto "`.`". Olhe mais uma vez os conteúdos do arquivo tar, usando a opção `-t`, para ver as diferenças em relação ao caso anterior.

6. Vamos agora extrair os conteúdos do arquivo tar. Para evitar problemas com os seus diretórios, crie um novo subdiretório em sua conta, dentro do qual vamos vamos fazer nossas tentativas, chamando-o por exemplo de `test`. Entre neste diretório e execute o comando

   ```
   tar -xvf ../my-web-hp.tar
   ```

   Note a chave `-x`, que é uma instrução para extrair arquivos de dentro do arquivo tar. Observe que não há argumento de linha de comando, o que significa que todos os arquivos e diretórios existentes no tar serão extraídos. A chave `-v`, como sempre, faz com que o comando `tar` nos diga o que está fazendo.

7. Podemos extrair de dentro do arquivo tar apenas parte de seus conteúdos, até um único arquivo ou diretório. No caso de diretórios, a ação é sempre recursiva dentro deles. Para fazer isto, continue no diretório `teste` e remova todos os arquivos dentro dele. Depois use o `tar` com a opção `-t` para listar os conteúdos do arquivo `../my-web-hp.tar`. Usando o nome de um dos arquivos, extraia um único arquivo do tar usando o comando

   ```
   tar -xvf ../my-web-hp.tar <nome-do-arquivo>
   ```

Em geral é necessário usar o nome completo do arquivo, ou seja, o path completo exatamente como ele aparece dentro do tar, mas em nosso caso aqui há apenas nomes simples. Se o nome for um path contendo diretórios, estes diretórios serão criados como subdiretórios do diretório corrente.

8. É possível acrescentar um arquivo adicional a um tar que já existe. Como aconteceria numa fita, ele será colocado em sequência no final do arquivo. Acrescente ao tar o arquivo que acaba de extrair dele, usando

   ```
   tar -rvf ../my-web-hp.tar <nome-do-arquivo>
   ```

   Observe que, quando o arquivo foi extraído do tar, ele não foi apagado deste e sim simplesmente copiado. Assim, teremos agora duas cópias do arquivo que acabamos de colocar dentro tar, com o mesmo nome. Isto é possível porque, como seria típico para uma fita, o tar trata os seus conteúdos de forma simplesmente sequencial. Verifique os conteúdos do arquivo tar usando a opção `-t`. Ao se extrair arquivos do arquivo tar, eles serão sempre extraídos na ordem que se encontram ao longo dele. Assim, se extrairmos todos os arquivos deste tar, a segunda cópia do arquivo sempre sobrescreverá a primeira. Por outro lado, se extrairmos apenas o arquivo em questão, ficaremos com a primeira cópia. Obviamente, não é uma idéia muito boa termos vários arquivos com o mesmo nome dentro de um arquivo tar.

9. Ao se criar um arquivo tar, é possível comprimir automaticamente o seu conteúdo usando o compressor `gzip`. Neste caso, em vez de criar uma pipeline explicitamente com a shell, o programa é capaz de criar esta pipeline para nós, internamente. Entre de novo no diretório `WWW` de sua conta e crie um novo arquivo tar, desta vez comprimido, usando

   ```
   tar -czvf ../my-web-hp.tgz .
   ```

   A opção `-z` instrui o `tar` a comprimir o conteúdo do arquivo tar que vai ser criado, usando para isto o comando `gzip`. A terminação `.tgz` é o padrão usual para arquivos tar comprimidos. Para olhar os conteúdos ou extrair arquivos de um arquivo tar comprimido, é necessário usar sempre a opção `-z`, instruindo desta forma o comando `tar` a descomprimir o conteúdo durante a leitura.

10. Para entender melhor o que realmente aconteceu na tarefa anterior,

vamos repeti-la de outra forma, usando uma pipeline explícita e o comando `gzip`. Ainda dentro do diretório `WWW`, execute a pipeline

```
tar -cvf - .  | gzip -c - >!  ../my-web-hp.tgz
```

É necessário explicar o que está acontecendo em cada uma das partes deste comando. O comando `tar` está sendo usado para criar um arquivo tar, sem compressão. O argumento `-` da opção `-f` instrui o `tar` a enviar o resultado de sua ação para o canal padrão de saída "stdout", em vez de para um arquivo em disco. O comando `gzip` com o argumento `-` lê dados do canal padrão de entrada "stdin" e os comprime, enquanto a opção `-c` o instrui a enviar o resultado para stdout. Por fim, o redirecionador `>!` envia os dados que aparecem no stdout do `gzip` para o arquivo mostrado, sobrescrevendo-o se necessário. Como sempre, o sinal de pipeline `|` envia o stdout de um programa para o stdin do próximo.

Na tarefa anterior toda esta operação foi feita internamento pelo próprio comando tar, devido ao uso da opção `-z`. Verifique o conteúdo do arquivo que acabamos de sobrescrever, usando mais uma vez o `tar` com as opções `-tzvf`.

11. A idéia, usada na tarefa anterior, de se criar um arquivo tar que é enviado para stdout é de extraordinária utilidade. Com ela, podemos usar o `tar` em pipelines para movimentar árvores completas de diretórios e arquivos de um lado para outro dentro do sistema. Para ilustrar como fazer isto, primeiro remova todos os arquivos que há dentro do diretório `teste` e vá para a raiz de sua conta. Uma vez lá, você pode transferir todos os conteúdos do diretório `WWW` para o diretório `teste` usando a pipeline

    ```
    tar -C WWW -cf - .  | tar -C teste -xvf -
    ```

    Tome cuidado para não errar os detalhes, pois o arquivo tar que está sendo passado pela pipeline precisa ser criado em `WWW` e extraído em `teste`, não o contrário.

    A opção `-C` instrui o `tar` a mudar internamente de diretório antes de criar ou ler qualquer coisa. Assim, o primeiro `tar` está entrando dentro de `WWW` para criar o arquivo tar, que está sendo enviado para stdout. O segundo `tar` está entrando em `teste` antes de ler os dados de stdin, o que é feito devido ao uso do argumento `-` da opção `-f` junto com a opção `-x`. Note que só colocamos a opção `-v` em um dos dois comandos pois, de outra forma, os outputs para a tela dos dois comandos `tar` que estão rodando ao mesmo tempo na pipeline se misturariam na tela, causando confusão.

Note que uso da descrição "arquivo tar" é um abuso de linguagem neste caso, pois o arquivo em si nunca chega a existir. O que há é um stream de dados que está sendo passado entre os dois processos, o da primeira instância do comando `tar` e o da segunda instância. O formato deste stream de dados é o formato de um arquivo tar, ou seja, trata-se do mesmo conteúdo byte por byte, mas não há um arquivo do filesystem envolvido nisto. Trata-se aqui de uma ferramenta muito ágil de transferência de dados entre dois diretórios ou mesmo entre dois filesystems.

12. Além de escrever arquivos tar em disco e passar streams de tar por pipelines, podemos escrever arquivos tar em dispositivos tais como disquetes. Não se trata aqui de criar um filesystem no disquete e colocar um arquivo tar nele, o que certamente é possível mas não seria diferente do que já vimos em aulas anteriores. Trata-se, isto sim, de usar diretamente todo o espaço disponível no floppy para colocar o arquivo tar, no estilo do que seria feito com uma fita magnética. Combinado com compressão, este método é a forma mais eficiente de se usar o espaço disponível no floppy para o armazenamento e transporte de dados.

    Para fazer isto, basta usar como argumento da opção `-f` o dispositivo `/dev/floppy` que corresponde ao floppy. Coloque um floppy vazio no drive, pois quaisquer conteúdos que existam nele seriam perdidos, entre no diretório `WWW` e execute

    ```
    tar -cvf /dev/floppy .
    ```

    Observe que, quando usado desta forma, sem filesystem, o floppy perde a sua característica de dispositivo de acesso randômico e passa a se comportar como um dispositivo de acesso sequencial, de forma semelhante a uma fita magnética. A única diferença é que não é necessário rebobinar o floppy! Para verificar o conteúdo do floppy basta usar o comando

    ```
    tar -tvf /dev/floppy
    ```

13. Para guardar no floppy mais do que os 1440 blocos de 1024 bytes (ou seja, aproximadamente 1.4 MB) que constituem o espaço físico disponível em um floppy comum, pode-se usar compressão. Dependendo do tipo de dados que se quer guardar, isto pode dobrar o espaço efetivamente disponível, ou até mais o que isto. Para fazer isto basta acrescentar a chave `-z`, como já vimos antes,

    `tar -czvf /dev/floppy .`

    Note que, ao se fazer isto, os conteúdos anteriores do floppy foram sobrescritos sem aviso e perdidos, inclusive a estrutura de filesystem, se havia alguma nele. Para listar ou extrair os conteúdos do floppy é necessário usar, mais uma ver, a opção `-z`, como em

    `tar -tzvf /dev/floppy`

    Observe que, como não existe estrutura de filesystem no floppy escrito desta maneira, nem nomes de arquivos envolvidos, é importante que se registre na etiqueta do floppy a natureza do seu conteúdo.

14. Caso se queira fazer em floppies um arquivo tar maior do que 1.4 MB, é necessário usar a capacidade do comando `tar` de criar um arquivo tar em múltiplos volumes. Para fazer isto basta usar a opção `-M` do comando. Obtenha vários disquetes vazios e tente fazer um backup multi-volume de sua conta inteira neles, indo para a raiz, colocando o primeiro disquete do drive e usando o comando

    `tar -cvMf /dev/floppy .`

    Se a sua conta contiver mais de 1.4 MB, o que é provável, o `tar` vai escrever o primeiro floppy até o fim e em seguida dar uma pausa e pedir para você colocar no drive o segundo volume, ou seja, o segundo disquete. Isto se repetirá a cada disquete que for necessário até que todos os dados tenham sido guardados. Não há limite para o número de disquetes que você pode usar em sequência desta maneira, exceto pela sua paciência de ficar trocando os floppies no drive.

15. Observe que, ao escrever sequencialmente os disquetes, é muito importante numerá-los, além de marcar em cada um a natureza do seu conteúdo, pois eles precisam ser usados na ordem certa para a leitura. Caso trate-se realmente de um backup de sua conta, é também uma boa idéia proteger todos os disquetes contra escrita, depois de terminar de

escrevê-los. Para listar os conteúdos ou extrair dados do conjunto dos disquetes, basta usar a mesma chave `-M`. Por exemplo, liste os conteúdos do seu backup colocando o primeiro disquete no drive e executando

```
tar -tvMf /dev/floppy
```

Da mesma forma que antes, os disquetes subsequentes serão solicitados pelo programa conforme se tornarem necessários. Observe que não é possível fazer um backup multi-volume usando compressão. Para conseguir combinar estas duas coisas é necessário usar um truque, como será proposto nos problemas ao final desta apostila.

Note que não estamos recomendando aqui backups em floppies como a melhor forma de preservar os dados de sua conta. A não ser que você adquira floppies de qualidade excepcional, trata-se de uma mídia muito propensa a erros e de baixa confiabilidade. É muito melhor fazer backups em ZIPs. Neles é possível simplesmente criar filesystems e copiar diretamente os arquivos, ou fazer um backup sem filesystem como descrito aqui, inclusive com vários volumes, usando o dispositivo `/dev/zip`.

Usando-se compressão pode-se colocar grandes quantidades de dados em um único floppy ZIP de 100 MB, trata-se de uma mídia muito mais confiável e, considerando-se o preço por unidade de armazenamento, eles podem até ser de fato efetivamente mais baratos do que floppies comuns! Num futuro próximo a melhor forma de fazer backups dos dados de sua conta será em discos óticos CDRW mas por ora, como o uso deste tipo de disco ainda é operacionalmente um pouco complexo, a melhor recomendação para backups ainda é o disco ZIP padrão de 100 MB.

# 3 Problemas e Dicas

1. Como vimos em uma das tarefas, é possível colocar vários arquivos com exatamente o mesmo nome dentro de um arquivo tar. Ao se extrair arquivos deste arquivo tar, eles serão sempre extraídos na ordem que se encontram ao longo dele, o que implica que o último arquivo será o único sobrevivente. Obviamente, não é uma idéia muito boa termos arquivos com o mesmo nome dentro de um arquivo tar. Entretanto, existe uma forma de colocar um novo arquivo dentro do tar de tal forma

que ele substitua o outro de mesmo nome que já está lá, atualizando desta forma o arquivo tar. Olhe as páginas de manual do `tar` para aprender como usar as opções `--delete` e `-r` para fazer isto.

2. Para fazer um backup multi-volume usando compressão, é necessário primeiro fazer um arquivo tar comprimido em disco e depois usar o `tar` com a opção multi-volume para fazer um tar deste tar, por exemplo em disquetes! Tente fazer isto com o conteúdo de algum diretório apropriado de sua conta. Isto pode ser útil para resolver problemas pontuais tais como transportar grandes quantidades de dados para um computador que não esteja na rede e que disponha apenas de uma unidade floppy como dispositivo de entrada e saída. Não se trata, naturalmente, de uma boa forma de se fazer backups regulares de sua conta.

3. Considere também a seguinte alternativa para fazer o tipo de operação descrito no ítem anterior: fazer o arquivo tar comprimido em disco, depois dividi-lo em vários blocos consecutivos de exatamente 1440 KBytes cada um e finalmente colocar cada bloco num disquete. Você vai ter de ler as páginas de manual e aprender a usar os comandos `split`, que serve para fazer a divisão do arquivo em blocos, e `dd`, que serve para copiar em nível baixo (sem filesystem) os blocos para os disquetes. Depois é só inverter as operações e usar o comando `cat` (de conCATenate) para reconstituir o arquivo tar comprimido.

4. Conceba e implemente um sistema de backup rotineiro para a sua conta, escolhendo entre as várias alternativas disponíveis: floppies ou ZIPs; com ou sem filesystem; usando `cp` ou `tar`; usando ou não programas como o que desenvolvemos anteriormente para este tipo de finalidade; single-volume ou multi-volume. Olhe as páginas de manual do `tar` para informar-se sobre as alternativas existentes para se fazer backups incrementais. Preocupe-se tanto com os aspectos de praticidade quanto com os de segurança dos dados. E, uma vez feita a sua escolha, considere passar a usar este sistema regularmente para salvar e guardar os dados de sua conta. Apesar de que hoje temos backup de contas de usuários no projeto Sócrates, é sempre conveniente você ter o seu próprio backup, que também poderá usar para, por exemplo, transportar seus dados do Instituto para casa e vice versa.

5. Com o uso do editor `emacs` é possível examinar e manipular o conteúdo

de um arquivo tar, tratando-o como se fosse um diretório. Isto inclui o uso do editor para modificar os arquivos contidos no arquivo tar! Pode-se fazer isto até com arquivos tar comprimidos! Explore as possibilidades deste uso avançado do `emacs`. Basta criar um arquivo tar comprimido tal como o `my-web-hp.tgz` que criamos durante as tarefas e simplesmente editá-lo com o `emacs`! Você estará usando o modo "dired" do `emacs`, que permite a manipulação de diretórios de arquivos, o que pode ser novidade para você. Mas não é muito difícil achar o caminho das pedras com um pouco de tentativa e erro.

6. Considere fazer um sistema de backup de sua conta em discos CDRW em formato ext2. A idéia básica é primeiro escrever o backup numa imagem do filesystem criada dentro de um arquivo e montada por loopback, depois escrever a imagem na mídia usando o comando apropriado. Para ler e recuperar os seus dados você poderá usar diretamente uma unidade CDROM qualquer, por exemplo na sala Pró-Aluno, onde todas as máquinas têm estas unidades e onde está configurada uma montagem de usuário para CDROMs em formato ext2. Você terá de rever e adaptar o método que foi exposto na apostila anterior para imagens de uma partição de ZIP. O número de blocos de 1024 bytes com que o arquivo para a imagem deve ser criada é, neste caso, 663328. Há no diretório `/temp` do sistema popio da Pró-Aluno um arquivo `README.EXT2CD` com algumas instruções úteis.

# 4 Uma Lista de Comandos Relevantes

`gzip [-c] [<arquivo>]` : comprime o arquivo, acrescentando ao seu nome a extensão `.gz`; sem argumento, lê dados de stdin; com a opção `-c`, envia o output para stdout.

`tar -|c|t|x|r| [-vz] -f <arquivo> <conteúdos>` : cria (`-c`), lista conteúdos (`-t`), extrai arquivos (`-x`) ou acrescenta arquivos (`-r`) em um arquivo tar, com os conteúdos dados como argumentos; a opção `-f` determina qual é o arquivo tar; a opção `-v` faz o programa relatar o que está fazendo; a opção `-z` causa a compressão dos dados escritos e a descompressão dos dados lidos; há `muito` mais opções do que estas, veja a página de manual.