FMA 215

Aula 3: Um Passeio Pelo Sistema

Prof. Jorge L. deLyra

01 de Abril de 2002

1 Elementos Sintáticos destas Notas

Nestas notas <comando> deve sempre ser substituído pelo nome de algum comando em particular (sem os sinais de < e >). A mesma convenção vale para <arquivo>, <diretório>, etc. A convenção <algum-tipo-de-coisa> é usada para indicar que neste ponto deve-se substituir um exemplo do tipo de coisa indicado.

Todos os comandos, arquivos e demais coisas que estariam sendo digitadas ao terminal são mostradas num `tipo de letra de largura fixa`. Teclas que você deve digitar serão representadas pelo seu nome entre brackets quadrados como, por exemplo, `[Ctrl]` e `[Alt]`. Estas são partes de um conjunto de convenções usadas de forma geral em documentos sobre o sistema. Não há regras muito fixas, mas uma certa forma relativamente padronizada de se fazer as coisas.

Dada a sua utilização constante, muitos substantivos do Inglês serão usados sem tradução ou aviso. Às vezes faremos o mesmo com verbos também, atribuindo a eles alguma conjugação arbitrária do Português (eu booto, você boota, eles bootam, etc). Aos que se ofenderem com isto, sugiro uma procura consistente por equivalentes destas palavras no Tupi que, segundo alguns, é a verdadeira língua original deste país `|:-)`.

2 Alguns Elementos Sintáticos Relevantes

A sintaxe básica do sistema é relativamente simples, mas com o tempo vai se complicando e torna-se vasta. Aplicativos e sub-sistemas podem ter as suas próprias sintaxes. Há uma espécie de filosofia geral de como as coisas devem ser, mas não um conjunto muito rígido de regras, pois uma das idéias centrais da coisa toda é manter uma grande flexibilidade. Por enquanto, a pequena lista que segue basta para as coisas mais básicas que precisamos saber de imediato.

Execução de comandos: para executar um comando, digita-se o seu nome ao terminal, terminando com a tecla "Enter" ou "Return" do teclado. Não há diferença de procedimento para comandos intrínsecos da shell, demais comandos do sistema, programas e aplicativos. Faz-se sempre a mesma coisa para se iniciar um destes programas ou aplicativos, apesar de que o comportamento de cada um pode ser bem específico e muito diferente do comportamento dos outros.

A diferença entre "programas", "comandos" e "aplicativos" é apenas convencional, todos são programas que o sistema roda. Em geral chamamos de comandos programas relativamente pequenos que fazem uma coisa definida e terminam. São chamados de aplicativos programas maiores, que podem entrar em um relacionamento interativo com o usuário, aceitando dele um conjunto de comandos próprios, internos àquele programa em particular.

`<comando>` roda o programa e coloca o resultado na tela, que é o comportamento mais comum dos programas. Muitos programas aceitam ou requerem argumentos, muitas vezes nomes de arquivos.

`<comando> > <arquivo>` redireciona o output do comando para dentro de um arquivo, criando o arquivo. É necessário colocar o nome de algum arquivo que ainda não exista no diretório onde se está.

`<comando> >> <arquivo>` redireciona o output do comando para dentro de um arquivo já existente, colocando-o ao final do arquivo, depois do conteúdo que já estiver lá.

Nomes de arquivos: o nome completo de um arquivo consiste da sequência de diretórios, sendo cada um deles um sub-diretório do anterior, que é necessário percorrer, a partir da raiz do sistema de arquivos, para chegar até ele. Os diversos diretórios são separados por caracteres `/` e, ao final, temos o nome do arquivo em si. Arquivos também têm nomes relativos, trata-se da mesma coisa mas começando de algum outro ponto do sistema de arquivos.

/<diretório>/···/<diretório>/<arquivo> nome ou path absoluto de um arquivo, ou seja, começando na raiz do sistema de arquivos.

<diretório>/···/<diretório>/<arquivo> nome ou path de um arquivo relativamente ao diretório onde você está, ou seja, começando num subdiretório do diretório onde você está.

3 Tarefas

Dê um passeio pelo sistema, incluindo pelo menos as tarefas contidas na lista abaixo. Para descobrir como realizar algumas delas, examine a lista de comandos que existe ao final desta apostila. Anote suas dúvidas, dificuldades e curiosidades para esclarecimento na aula seguinte. Para cada um dos comandos ou procedimentos, examine cuidadosamente o output e veja se dá para entender o que ele significa. Se não der, então você já tem algum material para a sua lista de dúvidas.

Lembre-se de que, assim que entrar no sistema, você estará em um diretório particular que é a sua “conta” ou área de disco no sistema. Nós o chamamos de seu “home” ou de “raiz da sua conta”. Em geral ele será o diretório /home/<seu-username>/. Esta é a área do sistema onde você pode escrever, criar arquivos. A maior parte do resto do sistema está acessível para leitura por você, mas não para escrita, ou seja, você pode ler e copiar arquivos de lá, mas não pode apagar arquivos ou criar novos arquivos nestas áreas.

Estas tarefas podem ser realizadas tanto dentro do sistema de janelas X11 quanto em um terminal de caracteres. Se você já está habituado/a com os sistemas de janelas, use o X11, caso contrário use um console ou terminal de caracteres, para evitar confusão. Você pode ir para um terminal de console acionando simultaneamente as teclas `[Ctrl]`, `[Alt]` e `[F1]`. Para voltar para o X11 digite `[Ctrl]`, `[Alt]` e `[F7]`, mas não se esqueça de sair de sua sessão ou sessões antes de abandonar o terminal.

1. Faça um login no sistema, usando seu “username” e sua “password”. Lembre-se de nunca usar os keypads para digitar a sua senha.

2. Mostre a data e a hora do sistema, usando o comando `date`.

3. Verifique quem mais está usando o sistema além de você, usando o comando `w`.

4. Verifique que processos você está rodando, usando o comando `ps`.
5. Verifique que filesystems estão disponíveis no sistema, usando o comando `df`.
6. Liste todas as variáveis definidas na sua shell, usando o comando `set`.
7. Liste todas as variáveis existentes no seu environment, usando o comando `env`.
8. Limpe a tela do seu terminal, usando o comando `clear`.
9. Liste os arquivos que existem em sua conta, usando o comando `ls`.
10. Crie um arquivo novo em sua conta, chamado `trash`, usando o comando `touch`.
11. Liste os arquivos do seu diretório-home para verificar que o novo arquivo está lá.
12. Crie um diretório novo, chamado `tmp`, usando o comando `mkdir`; este será um sub-diretório de seu diretório-home.
13. Liste os arquivos do diretório para verificar que o novo diretório está lá.
14. Movimente o arquivo `trash` para dentro do diretório `tmp`, usando o comando `mv`.
15. Vá para dentro do sub-diretório `tmp` de sua conta, usando o comando `cd`.
16. Ecoe a frase "Hello world" na tela do terminal, usando o comando `echo`.
17. Escreva esta mesma frase dentro do arquivo `trash`. Dica: use redirecionamento de output, como explicado na seção 2 desta apostila.
18. Mostre o conteúdo do arquivo `trash` na tela, verificando que o texto está lá, usando o comando `cat`.
19. Mostre uma lista dos arquivos existentes no diretório-raiz do sistema de arquivos (`/`).

20. Acrescente esta lista ao final do arquivo `trash`. Dica: use mais uma vez redirecionamento de output.
21. Mostre o conteúdo do arquivo `trash` na tela, verificando que a lista de fato está lá.
22. Copie o arquivo `trash` com o nome `trash2`, usando o comando `cp`.
23. Liste os arquivos do diretório para verificar que o novo arquivo está lá.
24. Mostre o tipo de arquivo de cada um dos dois arquivos, usando o comando `file`.
25. Remova o arquivo `trash`, usando o comando `rm`.
26. Mude o nome do arquivo `trash2` para `trash`, usando o comando `mv`.
27. Crie um novo link para o arquivo `trash`, chamado trash2, usando o comando `ln`.
28. Liste os arquivos do diretório para verificar que o novo arquivo está lá.
29. Mostre o tipo de arquivo de cada um dos dois arquivos.
30. Remova o diretório `tmp` que você criou, bem como os arquivos dentro dele.
31. Limpe a tela do seu terminal.
32. Vá para a raiz do sistema de arquivos (o diretório `/`).
33. Dê uma boa olhada dentro de alguns dos sub-diretórios que existem lá; em particular, olhe os diretórios `etc`, `bin`, `lib`, `dev`, `var` e `usr`.
34. Determine qual é o tipo de arquivo mais comum em cada um dos sub-diretórios `etc`, `bin`, `lib`, `dev`, `var` e `usr`.
35. Limpe a tela do seu terminal.
36. Faça um logout do sistema, ou seja, termine a sua shell, usando os comandos `exit` ou `logout`.

4 Problemas e Dicas

1. Você saberia como determinar o número total de arquivos existentes neste sistema? E o número total de arquivos de cada tipo? Tente fazer isto, usando os comandos `ls`, `wc` e `find`. Você vai ter de usar várias das opções existentes nestes comando como, por exemplo, as opções `-l` e `-R` do comando `ls`. Dê uma boa lida nas páginas de manual de cada um destes comandos, executando `man ls` e assim por diante. Você pode andar para cima e para baixo dentro do `man` usando as teclas `[PageDown]` e `[PageUp]`. Para sair do `man`, digite a letra `q`.

5 Uma Lista de Comandos Relevantes

Os comandos são mostrados em ordem alfabética. Cada um é acompanhado por uma pequena explicação da sua função quando usado da forma mostrada. A lista e as explicações estão longe de esgotar todas as possibilidades, trata-se apenas de alguns exemplos muito particulares, mas que são muito básicos e importantes.

Cada um destes comandos aceita, além dos argumentos mostrados, uma série de opções diferentes, que podem mudar o seu comportamento. Estas opções são, em geral, letras individuais precedidas de um sinal "`-`", tais como `-a`, `-f`, `-k`, `-l`, `-n`, `-s`, etc. Mais tarde veremos várias delas, pois há algumas que são de fato extremamente úteis.

Observe que o comando `man` pode ser usado para obter mais informação (em Inglês) sobre os outros comandos. Para sua conveniência de uso, esta lista está disponível em uma página separada, encontrando-se ao final da apostila.

`cat <arquivo>` : mostra o conteúdo do arquivo.

`cd <diretório>` : muda para o diretório; sem argumentos, vai para o diretório-home do usuário.

`clear` : limpa a tela.

`cp <arquivo> <novo-arquivo>` : copia o arquivo, criando um novo arquivo.

`date` : mostra a data e a hora.

`df` : mostra os filesystems disponíveis, ou seja, que estão montados no sistema.

echo <string> : ecoa o string de caracteres na tela; é um comando built-in da shell.

env : mostra as variáveis de ambiente da sua sessão; é um comando built-in da shell.

exit : termina a sessão, ou seja, a shell; é um comando built-in da shell.

file <arquivo> : mostra o tipo do arquivo.

history : mostra a lista de comandos executados; é um comando built-in da shell.

ln [-s] <arquivo> <novo-nome> : sem opções, faz um novo "hard link" para o arquivo, ou seja, define um novo nome para ele na estrutura do filesystem; com a opção -s faz um novo "soft link" para o arquivo, ou seja, define um novo nome para ele, criando um novo arquivo que "aponta" para o antigo.

ls [-a] [-l] [<diretório ou arquivo>] : lista o arquivo ou os arquivos do diretório; sem argumentos, usa o diretório onde você está; com a opção -a mostra todos os arquivos do diretório, incluindo os arquivos especiais e demais arquivos "escondidos"; com a opção -l mostra informações mais completas sobre o arquivo ou arquivos, incluindo o tipo de arquivo (em termos da estrutura do filesystem, não do tipo de conteúdo), controle de acesso, número de links, ownership, tamanho, data e nome do arquivo; as ações das chaves -a e -l podem ser combinadas; aqui está um exemplo:

```
-rw-rw-r--  1 delyra  delyra  12 May  1 11:54   teste.txt
tuuugggooo  l user    group   s  time_and_date  name
```

man <nome-de-comando> : mostra a página do manual do comando; para sair, basta digitar a letra q (de "quit").

mkdir <diretório> : se o nome dado como argumento não contém o caracter /, cria um diretório que passa a ser um sub-diretório do diretório onde você está.

mv <arquivo> <novo-nome ou diretório> : muda o nome de um arquivo ou movimenta o arquivo para dentro do diretório.

`ps [a][u][x]` : mostra processos que estão rodando no sistema; sem opções, mostra apenas os processos do usuário que estão conectados a um terminal; a opção `a` inclui os processos de outros usuários; a opção `u` faz com que o comando mostre o usuário ao qual pertence cada processo; a opção `x` inclui os processos que não estejam conectados a um terminal.

`rm <arquivo>` : remove o arquivo.

`rmdir <diretório>` : deleta um diretório, desde que ele esteja vazio, ou seja, que não existam arquivos dentro dele.

`set [<variável>] [= (<valor>...<valor>)]` : sem argumentos, mostra todas as variáveis definidas na sua shell; com argumentos, atribui os valores a cada uma das palavras (words) da variável; se só houver um valor pode-se omitir os parêntesis; é um comando built-in da shell `tcsh`.

`touch <arquivo>` : cria um arquivo vazio.

`w` : mostra quem está logado no sistema.

6 Arquivos de Configuração

Alguns dos comandos utilizados nesta apostila têm o seu comportamento afetado pelos arquivos de configuração da "shell" que você está usando, que se chama `tcsh`. Estes arquivos são muito importantes para a configuração do seu ambiente de trabalho dentro do sistema, que chamamos de "ambiente de usuário". Eles serão examinados em maiores detalhes mais tarde. Por ora, apenas dê uma olhada neles. Tome muito cuidado para não danificá-los ou deletá-los, pois neste caso o seu ambiente de usuário deixará de funcionar e sua vida dentro do sistema ficará muito difícil, a não ser que você já seja um usuário muito experiente.

Há dois arquivos de configuração principais da `tcsh`, um chamado `.login`, que é executado uma única vez quando você entra no sistema, outro chamado `.cshrc` (ou `.tcshrc`) que é executado cada vez que uma nova shell é iniciada. O que você vê como resultado da execução do comando `env` está definido basicamente em `.login`, enquanto o que mostram os comandos `set` e `alias` está definido de forma geral em `.cshrc`.

Se você der uma olhada nos arquivos `.login` e `.cshrc` da sua conta, você verá que de fato eles estão praticamente vazios. Na verdade tudo que contêm são chamadas para outros arquivos que mantemos no sistema, os quais definem um ambiente de usuários padronizado. É desta forma que implementamos no sistema um ambiente padrão para os usuários, que pode facilmente ser mantido de forma centralizada, sem negar aos usuários a possibilidade de se desligarem deste sistema centralizado e de passar a definir e manter suas próprias configurações de ambiente de usuários. Mais tarde veremos como se pode fazer isto.

Uma Lista de Comandos Básicos

cat <arquivo> : mostra o conteúdo do arquivo.

cd <diretório> : muda para o diretório; sem argumentos, vai para o diretório-home do usuário.

clear : limpa a tela.

cp <arquivo> <novo-arquivo> : copia o arquivo, criando um novo arquivo.

date : mostra a data e a hora.

df : mostra os filesystems disponíveis, ou seja, que estão montados no sistema.

echo <string> : ecoa o string de caracteres na tela; é um comando built-in da shell.

env : mostra as variáveis de ambiente da sua sessão; é um comando built-in da shell.

exit : termina a sessão, ou seja, a shell; é um comando built-in da shell.

file <arquivo> : mostra o tipo do arquivo.

history : mostra a lista de comandos executados; é um comando built-in da shell.

ln [-s] <arquivo> <novo-nome> : sem opções, faz um novo "hard link" para o arquivo, ou seja, define um novo nome para ele na estrutura do filesystem; com a opção -s faz um novo "soft link" para o arquivo, ou seja, define um novo nome para ele, criando um novo arquivo que "aponta" para o antigo.

ls [-a] [-l] [<diretório ou arquivo>] : lista o arquivo ou os arquivos do diretório; sem argumentos, usa o diretório onde você está; com a opção -a mostra todos os arquivos do diretório, incluindo os arquivos especiais e demais arquivos "escondidos"; com a opção -l mostra informações mais completas sobre o arquivo ou arquivos, incluindo o tipo de arquivo (em termos da estrutura do filesystem, não do tipo de conteúdo), controle de acesso, número de links, ownership, tamanho, data e nome do arquivo; as ações das chaves -a e -l podem ser combinadas; aqui está um exemplo:

```
-rw-rw-r--  1 delyra  delyra  12 May  1 11:54   teste.txt
tuuugggooo  l user    group   s  time_and_date  name
```

man <nome-de-comando> : mostra a página do manual do comando; para sair, basta digitar a letra q (de "quit").

mkdir <diretório> : se o nome dado como argumento não contém o caracter /, cria um diretório que passa a ser um sub-diretório do diretório onde você está.

mv <arquivo> <novo-nome ou diretório> : muda o nome de um arquivo ou movimenta o arquivo para dentro do diretório.

ps [a][u][x] : mostra processos que estão rodando no sistema; sem opções, mostra apenas os processos do usuário que estão conectados a um terminal; a opção a inclui os processos de outros usuários; a opção u faz com que o comando mostre o usuário ao qual pertence cada processo; a opção x inclui os processos que não estejam conectados a um terminal.

rm <arquivo> : remove o arquivo.

rmdir <diretório> : deleta um diretório, desde que ele esteja vazio, ou seja, que não existam arquivos dentro dele.

set [<variável>] [= (<valor>...<valor>)] : sem argumentos, mostra todas as variáveis definidas na sua shell; com argumentos, atribui os valores a cada uma das palavras (words) da variável; se só houver um valor pode-se omitir os parêntesis; é um comando built-in da shell tcsh.

touch <arquivo> : cria um arquivo vazio.

w : mostra quem está logado no sistema.