

UFSC / CTC / INE
**Disciplina: Paradigmas de
Programação**
(Programação Funcional)

Curso de Ciências da Computação: INE5416
Prof. Dr. João Dovicchi*

1 Aula Prática 1 - Unix / Haskell

Nesta aula vamos compreender alguns conceitos do ambiente a ser utilizado nas aulas de Paradigmas de programação. Em primeiro lugar, vamos compreender alguns princípios básicos do sistema Unix e, posteriormente, alguns aplicativos para interpretar compilar programas em Haskell e compilar programas em C. (se voce já domina o sistema Unix, vá para o roteiro 2, neste mesmo documento).

Embora existam diversas interfaces de programação ou *Integrated Development Environment* (IDE), é fundamental que o aluno use um editor simples, uma vez que ele deve compreender aspectos fundamentais da estrutura da linguagem. Lembrem-se: este curso é de paradigmas de programação, não de programação em C ou em Haskell.

1.1 O sistema Unix

O sistema Unix, diferentemente do que muitos acreditam, tem todas as ferramentas de desenvolvimento e interface gráfica (anterior ao sistema de janelas do DOS, hoje conhecido como Microsoft Windows). Muito embora existam diversos GUI (*Graphical User Interface*), o conceito do Unix é bem diferente

*<http://www.inf.ufsc.br/~dovicchi> --- dovicchi@inf.ufsc.br

de outros sistemas operacionais. Enquanto o MS Windows é um sistema operado por disco (DOS), os sistemas Unix (FreeBSD, Sun OS, AIX, HPUX etc.) e os *Unices-like* (Linux, BeOS, Hurd, Inferno etc.) são baseados no paradigma preemptivo de paginação de memória.

O aluno deve saber como utilizar um terminal para se conectar a um sistema Unix disponível. Para as aulas práticas o aluno deve familiarizar-se com o ambiente do terminal do Unix (*Shell*), bem como com os comandos básicos para criar e editar arquivos de programa, manipular arquivos e diretórios, compilar programas em C e Haskell e executá-los.

1.1.1 A linha de comandos (bash)

O Bourne Again Shell (bash) é um dos mais poderosos interpretadores de comandos do terminal Unix. Geralmente, sistemas como Linux, FreeBSD, SunOS e outros implementam este tipo de interpretador por *default*.

Normalmente, o sinal de pronto (*prompt*) do bash é algo semelhante a:

```
bash-2.05b$
```

ele indica que o sistema está pronto, esperando a entrada de uma linha de comando. Nestes roteiros, o sinal de pronto será representado pelo sinal “\$”. No caso da máquina haskell, o *prompt* tem o nome do usuário e da máquina seguido do diretório corrente:

```
[jcd@haskell ~]$
```

onde o sinal de til representa o diretório \$HOME (digite “pwd” para ver o diretório corrente).

Uma das melhores facilidades do bash é o fato de ele completar a linha de comandos quando usamos a tecla <tab>. Por exemplo, ao digitarmos:

```
$ a<tab><tab>
```

o sistema retornará todos os comandos disponíveis no \$PATH do usuário que comecem com a letra “a”.

Para as atividades práticas o aluno deve conhecer alguns comandos básicos para lidar com arquivos e executar algumas tarefas. a tabela 1 apresenta alguns comandos básicos.

Um fato importante é que o Unix sempre considera que o comando deve ser executado. Raramente, o sistema pede a confirmação. Alguns comandos têm uma opção para pedir confirmação (p. ex. `rm -i` sempre pedirá confirmação antes de remover o arquivo).

Comando	Descrição (use <code>man <comando></code> para mais informações)
<code>cd</code>	Muda diretório.
<code>cp</code>	Copia arquivo.
<code>ls</code>	Lista os arquivos do diretório corrente.
<code>man</code>	exibe a página de manual.
<code>mkdir</code>	cria um diretório.
<code>mv</code>	Move (ou renomeia) arquivo.
<code>pwd</code>	exibe o path do diretório corrente.
<code>rm</code>	remove arquivo.

Tabela 1: Comandos básicos do Unix

1.1.2 Edição arquivos

Existem vários editores de texto no Unix. O mais usado, certamente é o Vi. Outros editores como o pico, mcedit, joe e emacs podem ser usados. O aluno deve escolher o que melhor lhe servir e utilizá-lo na edição de seus textos de programas.

Embora o Vi seja um tanto diferente de outros editores, ele é um dos mais poderosos e, além disso, é universalmente encontrado nas máquinas Unix. No entanto, os outros são mais amigáveis. O pico é o editor de textos do pine e o mcedit é o editor do “midnight commander”. O joe é um tipo de wordstar.

Entretanto, nada impede que os arquivos sejam editados no seu editor de preferência (inclusive em outro sistema) e depois copiado para a máquina onde serão feitas as práticas, desde que estes estejam no formato de texto puro. No entanto, é importante lembrar que os sistemas Unix não usam dois caracteres de fim de linha, mas apenas o caractere ASCII “10” (\x0A ou LF). Assim, ao tentarem abrir um arquivo texto editado no Unix em um sistema DOS, o arquivo poderá não estar corretamente formatado. O DOS utiliza dois caracteres ASCII como final de linha “10 13” (\x0A\x0D ou LF CR).

2 Roteiro 1

- O aluno deve acessar e aplicar, na prática, o que foi aprendido sobre o sistema operacional Unix.
 - Uso do comando `ls` para listar arquivos;
 - Criar, renomear e apagar diretórios; e

- Aplicar os comandos `cat`, `echo`, `tail`, `head` e `more`.
- Usar um editor de texto (`vi`, `pico`, `mcedit` etc.) para editar um texto e selecionar um de sua preferência.

3 Roteiro 2

- Acesse o Moodle e encontre os arquivos `.lex` e `.y`
- Copie-os para um diretório em sua área na `marite.inf.ufsc.br`
- Abra-os com um editor de texto, `cat` ou `more` e analise os conteúdos
- Tente entender como a sintaxe da linguagem C é especificada.

Obs.: Para tua conveniência resuma tuas observações em um relatório simples (em `txt`) e mantenha-o arquivado para entrega junto com o trabalho final.