



Curso de Programação C em Ambientes Linux – Aula 01

Centro de Engenharias da Mobilidade - UFSC

Professores Gian R. Berkenbrock e Giovani Gracioli

<http://www.lisha.ufsc.br/C+language+course+resources>



Apresentação

Curso de Programação C em Ambientes Linux

Aula 01 de 5 – 04/08/2014

Apresentação

- Introdução a Linguagem C e Linux
- Não é para ensinar lógica de programação
- Tópicos:
 - Introdução ao Linux
 - Ambiente de programação no Linux
 - Linguagem de Programação C
- Sala A112
- 4-8/08/2014 – 9 às 12hs
- Esperam-se compromisso e dedicação dos alunos

Conteúdo desta aula

- Introdução ao Linux
 - Estrutura de diretórios
 - Acessando diretórios e arquivos
 - Criação de diretórios e arquivos
 - Execução de programas
 - Permissões de arquivos e diretórios
- Ambiente de programação no Linux
 - Compilador GNU gcc
 - Criação e compilação do primeiro programa em C no Linux
 - Uso do gcc
 - Fases da compilação e execução de programas



Introdução ao Linux

Curso de Programação C em Ambientes Linux

Aula 01 de 5 – 04/08/2014

Linux

- Histórico
- Kernel: www.kernel.org – 3.16
- Algumas características:
 - É livre e desenvolvido voluntariamente por programadores experientes, hackers, e contribuidores espalhados ao redor do mundo
 - Recebe apoio de grandes empresas como IBM, HP, etc.
 - Multitarefa real
 - Multiusuário
 - Suporte a nomes extensos de arquivos e diretórios (255 caracteres)
 - Utiliza permissões de acesso a arquivos, diretórios e programas em execução na memória RAM.
 - Proteção entre processos executados na memória RAM
 - Suporte a mais de 63 terminais virtuais (consoles)

Comandos de ajuda

- `man [seção comando/arquivo]`
 - `man ls`, `man 5 hosts_access`
- `locate`
 - `locate bashrc`
- `which`
 - `which gcc`
- `find`
 - `find . -name "bashr*"`

Variáveis de ambiente

- Todo programa executa dentro de um ambiente e nele existem variáveis.
- Comandos:
 - export, echo, env
- Definidos em:
 - \$HOME/.bashrc /etc/profile /etc/bash.bashrc

Figure 9.2 Some important environment variables.

Variable name	Contains	Example
HOME	Your home directory	/home/larry
TERM	Your terminal type	xterm, vt100, or console
SHELL	The path to your shell	/bin/bash
USER	Your login name	larry
PATH	A list to search for programs	/bin:/usr/bin:/usr/local/bin:/usr/bin/X11

Sistema de Arquivos

- Tudo no linux é arquivo
- Arquivos são *Case Sensitive* – c/s extensão
- 255 caracteres
- Quando iniciados por “.” são ocultos
- Texto ou Binários
- Tamanho é medido em bytes:
 - Kbytes, Mbytes, Gbytes

Sistema de Arquivos - diretório

- *Case Sensitive*
- 255 caracteres
- Quando iniciados por “.” são ocultos
- Não podem possuir arquivos ou sub-diretórios com o mesmo nome
- Especificado por “/”
- /, /bin, /sbin, /usr, /usr/local, /mnt, /tmp, /var, /home, etc

Estrutura - diretório

O sistema GNU/Linux possui a seguinte estrutura básica de diretórios organizados segundo o FHS (Filesystem Hierarchy Standard):

- /bin - Contém arquivos programas do sistema que são usados com frequência pelos usuários.
- /boot - Contém arquivos necessários para a inicialização do sistema.
- /cdrom - Ponto de montagem da unidade de CD-ROM.
- /media - Ponto de montagem de dispositivos diversos do sistema (rede, pen-drives, CD-ROM em distribuições mais novas).
- /dev - Contém arquivos usados para acessar dispositivos (periféricos) existentes no computador.
- /etc - Arquivos de configuração de seu computador local.
- /home - Diretórios contendo os arquivos dos usuários.
- /lib - Bibliotecas compartilhadas pelos programas do sistema e módulos do kernel.
- /lost+found - Local para a gravação de arquivos/diretórios recuperados pelo utilitário fsck.ext2. Cada partição possui seu próprio diretório lost+found.

Estrutura – diretório 2

O sistema GNU/Linux possui a seguinte estrutura básica de diretórios organizados segundo o FHS (Filesystem Hierarchy Standard):

- /mnt - Ponto de montagem temporário.
- /proc - Sistema de arquivos do kernel. Este diretório não existe em seu disco.
- /sys - Sistema de arquivos do kernel. Este diretório não existe em seu disco.
- /root - Diretório do usuário root.
- /sbin - Diretório de programas usados pelo superusuário (root) para administração e controle do funcionamento do sistema.
- /tmp - Diretório para armazenamento de arquivos temporários criados por programas.
- /usr - Contém maior parte de seus programas. Normalmente acessível somente como leitura.
- /var - Contém maior parte dos arquivos que são gravados com frequência pelos programas do sistema, e-mails, spool de impressora, cache, etc.

Comandos - diretório

- `pwd` ou `“.”` – diretório atual
- `“..”` - diretório superior
- `“-”` - diretório anterior
- `cd` – mudança de diretório
 - Mudar o diretório padrão para `/etc` com o comando `cd /etc` e usar o comando `cat hosts`
- `tree [-L 1 -d]`
- `ls` – lista conteúdo do diretório
- `mkdir / rmdir`

Comandos - arquivos

- touch – cria vazio
- cat / tac – mostra conteúdo
- rm – remove
- cp – copia
- mv – move/renomeia
- tail/head – mostra fim de arquivo e começo
- ln - link

Redirecionamentos e Pipe

- > - redireciona a saída padrão para algum arquivo
- >> - adiciona ao final
- < - entrada padrão
- <<
 - cat << final
 - este arquivo
 - será mostrado
 - até que a palavra final seja
 - localizada no início da linha
 - final
- | - Envia a saída de um comando para a entrada do próximo comando para continuidade do processamento.

Permissões

- Tipo: donos, grupos e outros - **root**
- Acesso: leitura/listagem, escrita e execução/acesso
- `-rwxr-xr-- creidson users teste`
 - d ou l ou – seguido por 3 grupos de 3 letras
- `chmod`, `chgrp`, `chown`
- usuário (u), grupo (g), outros (o), todos (a).

Permissões especiais

- s
 - Dono - ajusta a identificação efetiva do usuário do processo durante a execução de um programa, também chamado de bit setuid. Não tem efeito em diretórios.
 - Grupo - ajusta a identificação efetiva do grupo do processo durante a execução de um programa, chamado de bit setgid. É identificado pela letra s no lugar da permissão de execução do grupo do arquivo/diretório. Em diretórios, força que os arquivos criados dentro dele pertençam ao mesmo grupo do diretório, ao invés do grupo primário que o usuário pertence.
 - S - Idêntico a "s".
- t - Salva a imagem do texto do programa no dispositivo swap, também chamado de stick bit.
 - Em diretórios, impede que outros usuários removam arquivos dos quais não são donos. Pode ser especificada somente no campo outros usuários das permissões de acesso.
 - T - Idêntico a "t".
- X - Se você usar X ao invés de x, a permissão de execução somente é aplicada se o arquivo já tiver permissões de execução. Em diretórios ela tem o mesmo efeito que a permissão de execução x.

Permissões especiais - 2

- Exemplo da permissão de acesso especial X:
- Crie um arquivo teste (digitando `touch teste`) e defina sua permissão para `rw-rw-r--` (`chmod ug=rw,o=r teste` ou `chmod 664 teste`).
- Agora use o comando `chmod a+X teste`
- digite `ls -l`
- Veja que as permissões do arquivo não foram afetadas.
- agora digite `chmod o+x teste`
- digite `ls -l`, você colocou a permissão de execução para os outros usuários.
- Agora use novamente o comando `chmod a+X teste`
- digite `ls -l`
- Veja que agora a permissão de execução foi concedida a todos os usuários, pois foi verificado que o arquivo era executável (tinha permissão de execução para outros usuários).
- Agora use o comando `chmod a-X teste`
- Ele também funcionará e removerá as permissões de execução de todos os usuários, porque o arquivo teste tem permissão de execução (confira digitando `ls -l`).
- Agora tente novamente o `chmod a+X teste`
- Você deve ter reparado que a permissão de acesso especial X é semelhante a x, mas somente faz efeito quando o arquivo já tem permissão de execução para o dono, grupo ou outros usuários.
- Em diretórios, a permissão de acesso especial X funciona da mesma forma que x, até mesmo se o diretório não tiver nenhuma permissão de acesso (x)

Permissões - octal

- 0 - Nenhuma permissão de acesso. Equivalente a -rwx.
- 1 - Permissão de execução (x).
- 2 - Permissão de gravação (w).
- 3 - Permissão de gravação e execução (wx). Equivalente a permissão 2+1
- 4 - Permissão de leitura (r).
- 5 - Permissão de leitura e execução (rx). Equivalente a permissão 4+1
- 6 - Permissão de leitura e gravação (rw). Equivalente a permissão 4+2
- 7 - Permissão de leitura, gravação e execução. Equivalente a +rwx (4+2+1)

Executando programas

- Precisa ter permissão
- Interno/Externo (PATH) – fora: ./comando
- Primeiro/Segundo Plano – fg/bg ou comando &
- Em sequencia : comando ; comando
- ps, top, kill

Pacotes

- instalando: `dpkg -i pct` ou `apt-get install pct`
- removendo `dpkg -r pct` ou `apt-get remove pct`
- Exemplo:

- `apt-get install gcc g++ make build-essential`

Compactação

- Zip
 - zip/unzip
 - zip textos.zip *.txt – unzip textos.zip
- Tar
 - tar xzvf arquivo.tar.gz – descompactar
 - tar czvf arquivot.tar.gz dir/* - compactar
 - Arquivos gerados pelo tar precisam ter a extensão .tar
 - Caso seja usada a opção -j para compactação, a extensão deverá ser .tar.bz2
 - Caso seja usada a opção -z para compactação, a extensão deverá ser .tar.gz ou .tgz



Programação no Linux

Curso de Programação C em Ambientes Linux

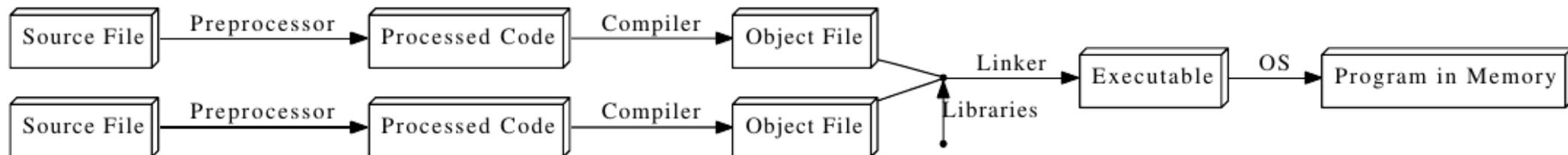
Aula 01 de 5 – 04/08/2014

Compilador gcc

- GCC – GNU Compiler Collection
- Instalar: pacotes gcc, g++ e bibliotecas se necessário
- Traduz código-fonte para código-alvo
- Suporta várias linguagens e processadores

Fases de compilação

- Pré-processamento: `gcc -E teste.c -o teste.i`
 - #arquivo de saída teste.i
- Compilação: `gcc -S teste.c`
 - #arquivo de saída teste.s em linguagem assembly da arquitetura alvo
- Montagem: `gcc -c teste.c`
 - #arquivo de saída teste.o - arquivo objeto
- C/M/L direta: `gcc -o teste teste.c`
 - # arquivo executável de saída teste



Fases de compilação 2

#Tipos de extensões de arquivo para o compilador gcc

#Extensão Interpretação

#.c Programa e linguagem C

#.C .cc Programa em linguagem C++

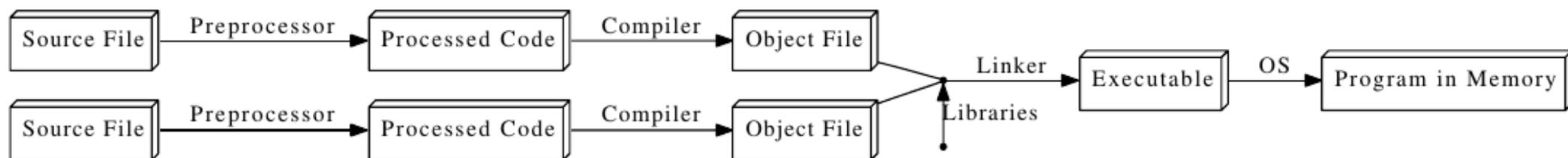
#.i Programa em C pré-processado

#.ii Programa em C++ pré-processado

#.S .s Programa em linguagem Assembly

#.o Programa objeto

#.a .so Bibliotecas compiladas



Estrutura básica de um programa em C

```
1  /* Figura 2.1: fig02_01.c
2     Primeiro programa em C */
3  #include <stdio.h>
4
5  /* função main inicia execução do programa */
6  int main( void )
7  {
8     printf( "Bem-vindo a C!\n" );
9
10     return 0; /* indica que o programa terminou com sucesso */
11 } /* fim da função main */
```

Bem-vindo a C!

Figura 2.1 ■ Primeiro programa em C.

Compilação

- Executar todos os comandos das fases
 - Criar o arquivo
 - Compilar (todos as fases)
 - Executar

Demo - GCC

```
#include <stdio.h>
```

```
double test_loop(double d, unsigned n)
```

```
{
```

```
    double x = 1.0;
```

```
    unsigned j;
```

```
    for (j = 1; j <= n; j++) x *= d;
```

```
    return x;
```

```
}
```

```
int main (void)
```

```
{
```

```
    double soma = 0.0;
```

```
    unsigned i;
```

```
    for (i = 1; i <= 1000000000; i++)
```

```
    {
```

```
        soma += test_loop(i, i % 5);
```

```
    }
```

```
    printf ("s = %g\n", soma);
```

```
    return 0;
```

```
}
```

Demo – GCC - 2

Primeiro teste :

```
gcc -Wall -O0 test.c -lm
```

```
time ./a.out
```

Segundo teste:

```
gcc -Wall -O1 test.c -lm
```

```
time ./a.out
```

Terceito teste:

```
gcc -Wall -O2 test.c -lm
```

```
time ./a.out
```

Quarto teste:

```
gcc -Wall -O3 test.c -lm
```

```
time ./a.out
```

Quinto teste:

```
gcc -Wall -O3 -funroll-loops test.c -lm
```

```
time ./a.out
```

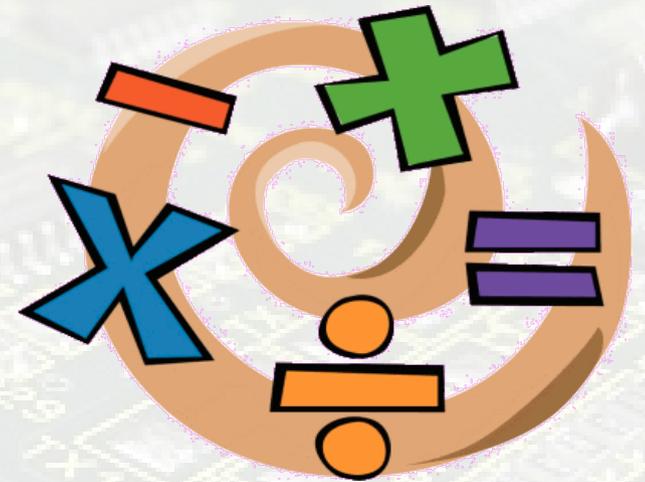
Sexto teste:

```
gcc -fdump-tree-<generic> test.c
```

Compilando um projeto

- Procurem um projeto escrito em C na página sf.net ou outro repositório que preferir
- Copiem ele localmente, descompactem, compilem e instalem
- Removam a instalação
 - Exemplo:
<http://sourceforge.net/projects/openprogrammer/files/OP/>

Finalizando



Sugestões Finais

- Resolvam a lista de exercícios relacionada com os temas da aula
 - Só se **aprende** a **programar**, **programando**
- Dúvidas sobre os exercícios podem ser enviadas por e-mail
- Leiam o material de apoio
 - Guia Foca – Iniciante: <http://www.guiafoca.org/cgs/guia/iniciante/index.html>
 - The Linux Documentation Project – Guides: <http://www.tldp.org/pub/Linux/docs/ldp-archived/users-guide/!INDEX.html>
 - Curso de C da UFMG: <http://mico.ead.cpdee.ufmg.br/cursos/C/>

Referências Bibliográficas

- Paul Deitel e Harvey Deitel, C: como programar, 6a edição, Ed. Prentice Hall Brasil, 2011.
- Curso de C da UFMG:
<http://mico.ead.cpdee.ufmg.br/cursos/C/>
- Guia Foca – Iniciante:
<http://www.guiafoca.org/cgs/guia/iniciante/index.html>
- The Linux Documentation Project –
Guides: <http://www.tldp.org/pub/Linux/docs/ldp-archived/users-guide/!INDEX.html>

