

Aula 17 - Parte 1

Bibliotecas Estáticas e Dinâmicas

MAC0216 - Técnicas de Programação I

Professores: Alfredo, Daniel, Fabio e Kelly

Departamento de Ciência da Computação
Instituto de Matemática e Estatística



Bibliotecas estáticas e dinâmicas

Bibliotecas estáticas

- ▶ São as bibliotecas cujo código é ligado ao programa invocador durante a construção (= *build*) do seu executável

Bibliotecas dinâmicas (ou compartilhadas)

- ▶ São as bibliotecas que são ligadas ao programa invocador depois que a execução do mesmo é iniciada

Ligação de bibliotecas dinâmicas

Pode ocorrer de duas maneiras:

- ▶ **Na inicialização**
 - quando o programa é carregado para execução, todas as bibliotecas dinâmicas que ele referencia também são carregadas com ele

- ▶ **Carga preguiçosa (mais comum)**
 - a biblioteca só é carregada quando ela for necessária para a continuidade da execução do programa

Bibliotecas dinâmicas

Vantagens

- ▷ Compartilhamento entre os vários programas que as usam (tanto na memória quanto no disco)
- ▷ Menor uso de memória na máquina como um todo (imagine o que aconteceria se a libc fosse estática...)
- ▷ Novas versões (com melhorias ou correções) das bibliotecas são aproveitadas mesmo pelos programas mais antigos

Bibliotecas dinâmicas

Desvantagens

- ▷ A carga de programas com bibliotecas dinâmicas pode ser mais lenta
- ▷ “DLL hell” - complicações frequentes relacionadas ao uso de bibliotecas dinâmicas nas versões mais antigas do Windows

Bibliotecas estáticas

Vantagens

- ▷ Não há o risco de não se localizar uma biblioteca em tempo de execução
 - o código das funções da biblioteca que um programa utiliza é adicionado ao seu executável final
- ▷ A versão das bibliotecas é fixa, portanto, não há o perigo de incompatibilidade de versões
- ▷ Mudanças no comportamento das bibliotecas novas não afetam a corretude dos programas antigos
- ▷ A carga de programas com bibliotecas estáticas é mais rápida

Bibliotecas estáticas

Desvantagens

- ▷ Tamanho do arquivo executável fica maior
- ▷ O código da biblioteca é adicionados ao código executável mesmo que em tempo de execução ele não seja usado

- ▷ Hoje em dia, bibliotecas dinâmicas são o padrão na maioria dos sistemas

Bibliotecas estáticas e dinâmicas no Linux

Como são nomeadas

- ▷ Bibliotecas estáticas: extensão **.a**
- ▷ Bibliotecas dinâmicas: extensão **.so**

Convenções

- ▷ Nomes de bibliotecas geralmente possuem o prefixo **lib**
 - Exemplos: `libteste.a` ou `libteste.so`

As bibliotecas de C usam essa convenção

- ▷ Exemplos: `libm.so`

Bibliotecas estáticas e dinâmicas no Linux

- ▷ No GCC, na ligação, a referência à biblioteca (na linha de comando) por meio do parâmetro **-l** não contém o prefixo e nem a extensão do nome da biblioteca
- ▷ Exemplo:

```
$ gcc meuprog.c -lm -lpthread
```

 - ▷ biblioteca matemática - /usr/lib/libm.so
 - ▷ biblioteca de threads - /usr/lib/libpthread.so

Criação e Ligação de Bibliotecas Estáticas

No Linux

Criação de uma biblioteca estática

1. Gerar os códigos-objeto dos fontes que farão parte da biblioteca estática
2. Executar o comando `ar` para compactar os objetos em um único arquivo com extensão `.a`

Exemplo

```
$ gcc -c calcmedia.c -o calcmedia.o -I/path/to/include
```

```
$ gcc -c calcdp.c -o calcdp.o -I/path/to/include
```

```
$ ar -rcv libestat.a calcmedia.o calcdp.o
```

- ▷ Para listar os objetos existentes dentro da biblioteca:

```
$ ar -t libestat.a
```

Ligação de uma biblioteca estática

- ▶ **Exemplo 1: biblioteca criada por um usuário**

```
$ gcc -o meuprog meuprog.c -static -lestat
```

ou, para indicar a localização da biblioteca, usar a opção **-L**

```
$ gcc -o meuprog meuprog.c -I/path/to/include  
-L/path/to/libs -static -lestat
```

- ▶ **Exemplo 2: biblioteca padrão de C**

Para forçar a ligação com a versão estática da biblioteca, usar opção **-static**

```
$ gcc -o meuprog meuprog.c -static -lm
```

Criação e Ligação de Bibliotecas Dinâmicas

No Linux

Criação de uma biblioteca dinâmica

1. Gerar os códigos-objeto dos fontes que farão parte da biblioteca dinâmica usando a opção **-fPIC** (para gerar um código independente de posição, ou seja, que funciona corretamente não importa em qual posição da memória ele for colocado)
2. Usar a opção **-shared** do GCC para gerar o arquivo **.so**

Exemplo

```
$ gcc -c -fPIC calcmedia.c -o calcmedia.o -I/path/to/include
```

```
$ gcc -c -fPIC calcdp.c -o calcdp.o -I/path/to/include
```

```
$ gcc -o libestat.so -shared calcmedia.o calcdp.o
```

Ligação de uma biblioteca dinâmica

Exemplo

```
$ gcc -o meuprog meuprog.c -I/path/to/include  
-L/path/to/libs -lestat
```

Observações:

- ▷ com a opção **-lestat** , o GCC busca a biblioteca de nome **libestat.so** (se fosse **-static -lestat** , procuraria **libestat.a**)
- ▷ a biblioteca não será incluída no executável **meu_prog**; ela será ligada dinamicamente ao programa na execução
- ▷ o comando **ldd** mostra as bibliotecas dinâmicas que serão carregadas para um executável. Ex.:

```
$ ldd meuprog
```

Execução de um programa que usa uma biblioteca dinâmica

Para que um executável encontre as bibliotecas para a ligação em tempo de execução, é preciso indicar ao ligador onde as bibliotecas devem ser procuradas. Isso pode ser feito de diferentes maneiras:

- ▷ Editando a variável de ambiente `LD_LIBRARY_PATH` (para uso temporário)

```
$ export LD_LIBRARY_PATH=/path/to/libs:$LD_LIBRARY_PATH
```

- ▷ Criando um arquivo `/etc/ld.so.conf.d/mylibs.conf` e incluindo nele o caminho

```
$ sudo echo "/path/to/libs" >> /etc/ld.so.conf.d/mylibs.conf
```

```
$ sudo ldconfig
```


Ligação de bibliotecas no Linux

Variáveis de ambiente:

- ▷ LD_LIBRARY_PATH - indica ao ligador de onde ele deve pegar as bibliotecas dinâmicas
- ▷ LD_PRELOAD - lista de bibliotecas a serem carregadas antes de todas as outras
- ▷ LD_BIND_NOW (*on* ou *off*)
 - *on* - faz as bibliotecas dinâmicas serem carregadas de uma vez no início da execução de um programa
 - *off* (padrão) - faz a carga sob demanda, i.e., *lazy binding*

Referências

- ▷ Notas de aula do prof. Fabio Kon

<https://www.ime.usp.br/~kon/MAC211/Aula9.html>

- ▷ Tutorial: *Static, Shared Dynamic and Loadable Linux Libraries*

<http://www.yolinux.com/TUTORIALS/LibraryArchives-StaticAndDynamic.html>