

19. - Biblioteca de funções

- Os arquivos .c estão associados aos headers correspondentes, logo tem-se que: `arquivo.h + arquivo.c = arquivo.o`
- Bibliotecas são arquivos que contêm módulos reutilizáveis pré-compilados que serão usados por desenvolvedores de aplicações.
- Bibliotecas:
 - possuem extensão .a
 - coleções de códigos objetos (extensão .o)
 - Exemplos:
 - libm.a**: contém o código objeto das funções declaradas em `math.h`
 - libX11.a** contém o código objeto das funções declaradas em `X11.h`
- Arquivos objetos: incluídos totalmente em um código executável.
- As funções das bibliotecas C padrão estão contidas em bibliotecas (se fossem em arquivos objetos, os programas seriam enormes).

Exemplos:

- LibC: biblioteca padrão C: funções básicas da linguagem C.
- GLib: funções de estrutura de dados
- NCurses, SDL, Allegro: construção de interfaces
- GDBM, SQLite: armazenamento em banco de dados

19.1 Construção de Bibliotecas

É possível desenvolver suas próprias bibliotecas, para usar em seus projetos ou disponibilizá-las a terceiros.

Bibliotecas podem ser: estáticas (static) ou compartilhada (shared).

- Bibliotecas estáticas fazem parte do corpo do programa principal, liberando-o assim do ambiente de configuração do sistema.
- Bibliotecas compartilhadas ligam-se ao programa principal dinamicamente, ou seja, um módulo só será ligado ao programa principal se for solicitado. Com isso, bibliotecas compartilhadas consomem menos recursos do sistema operacional além de facilitarem a substituição de módulos defeituosos sem a necessidade de compilar novamente todos os outros módulos ou sistemas envolvidos.
- `archive (ar)`: programa que junta vários arquivos objetos (.o) em uma biblioteca (.a)

Exemplos de parâmetros para o comando `ar`:

`r (replace)`: substituir versões anteriores dos arquivos na biblioteca, caso existam

`q (quick append)`: adiciona arquivos sem checar substituições

`c (create)`: cria arquivo se ele não existir, no caso de uma atualização ser requerida

`u (update)`: insere todos os arquivos listados e que sejam mais novos que os existentes.

`v (verbose)`: mostra na tela as inclusões que estão sendo realizadas

`s (symbols)`: cria uma tabela dos símbolos que estão sendo agregados à biblioteca

`t`: consultar o conteúdo de uma biblioteca

Bibliotecas estáticas

Exemplo1: cria uma biblioteca estática chamada libmods.a que inclui os dois arquivos objetos modulo1.o e modulo2.o

1. compilar os arquivos-fonte que irão fazer parte da biblioteca, e gerar os arquivos-objeto correspondentes.

```
> gcc -c modulo1.c modulo2.c
```

2. utilizar comando ar para juntar os arquivos e gerar a biblioteca libmods no diretório corrente

```
> ar rvs libmods.a modulo1.o modulo2.o
```

3. para consultar o conteúdo

```
> ar t libmods.a  
modulo1.o  
modulo2.o
```

ou outra opção:

```
> nm libmods.a
```

4. para incluir um terceiro módulo

```
> ar rvs libmods.a modulo1.o modulo2.o modulo3.o
```

5. para compilar o programa com sua biblioteca

```
> gcc seuProg.c -o seuExe libmods.a
```

outra opção é usar uma forma abreviada sem o lib e .a com -L. -L. faz com que o ligador inclua o diretório corrente (.) no path de bibliotecas, incluindo também somente os .objetos realmente necessários.

```
> gcc -L. seuProg.c -o seuExe -lmods
```

6. Se você deseja disponibilizar a biblioteca para outras pessoas, incluindo você mesmo, crie um diretório para guardar suas bibliotecas (por exemplo, /usr/local/lib) e copie a biblioteca gerada para lá. Para compilar programas use:

```
> gcc -L/usr/local/lib seuProg.c -o seuExe -lmods
```

o ligador irá procurar pela biblioteca nos diretórios default de bibliotecas (/lib, /usr/lib, /usr/local/lib, ...) e depois disso nos diretórios informados pela opção -L.

Biblioteca compartilhadas

Exemplo 2: criar uma biblioteca compartilhada "libmods.so"

1. compilar primeiro os códigos que irão compor a biblioteca com a opção `-fPIC` (*position independent code*)

```
> gcc -fPIC -c modulo1.c modulo2.c // até aqui a biblioteca é estática
```

2. criar a biblioteca dinâmica a partir dos códigos-objeto

```
> gcc -shared -o libmods.so modulo1.o modulo2.o  
> chmod og+rx libmods.so //se quiser que outros possam utilizar
```

3. compilar o programa com a biblioteca (similarmente às bibliotecas estáticas)

```
> gcc -L. seuProg.c -lmods
```

Se quiser crie um diretório para guardar suas bibliotecas (por exemplo, `/usr/local/lib`) e copie a biblioteca gerada para lá.

Para compilar programas use:

```
gcc -L/usr/local/lib seuProg.c -lmods
```

4. atualizar a variável de ambiente `LD_LIBRARY_PATH`, caso sua biblioteca estiver em um diretório não listado em `/etc/ld.so.conf`

```
> export LD_LIBRARY_PATH=.
```

```
> export LD_LIBRARY_PATH=/usr/local/lib
```

Veja que o nome da biblioteca se inicia com o prefixo lib.

O Linux usa o seguinte padrão para bibliotecas compartilhadas:

libaaa.so.b.c.ddd

libaaa.so é o nome da biblioteca;

b é um número que indicará a maior versão;

c é um número que indicará a menor versão;

ddd é um número que informa a "release".

Exemplo 3: utilizando makefile e bibliotecas

```
# autor: Prof. Armando N. Delgado
```

```
# nome da biblioteca, usado para gerar o arquivo libnome.a
```

```
LIBNOME = bib
```

```
# nome do arquivo header com as declarações da biblioteca
```

```
INCFILES = bib.h
```

```
# modulos que contêm as funções da biblioteca
```

```
LIBOBJ = modulo-1.o modulo-2.o modulo-3.o
```

```
# diretorio base onde estarão os diretórios de biblioteca
```

```
PREFIX = ${HOME}
```

```
# diretorio onde ficam os .h's de bibliotecas
```

```
INCDIR = $(PREFIX)/include
```

```
# diretorio onde ficam bibliotecas
```

```
LIBDIR = $(PREFIX)/lib
```

```
CC = gcc -g
```

AR = ar -rcu
INSTALL = install

.PHONY: clean distclean install all

%.o: %.c \$(INCFILES)
\$(CC) -c \$(CFLAGS) \$<

all: lib\$(LIBNOME).a

lib\$(LIBNOME).a: \$(LIBOBJ)
\$(AR) \$@ \$?
ranlib \$@

install: lib\$(LIBNOME).a
\$(INSTALL) ^ \$(LIBDIR)
\$(INSTALL) \$(LIBNOME).h \$(INCDIR)
@chmod 0644 \$(LIBDIR)/lib\$(LIBNOME).a \$(INCDIR)/\$(LIBNOME).h
@echo
@echo "*****"
@echo "A biblioteca \$(LIBNOME) está instalada no"
@echo "diretório \"\$(LIBDIR)\"."
@echo "O header \"\$(LIBNOME).h\" está no"
@echo "diretório \"\$(INCDIR)\"."
@echo "Para usar a biblioteca em compilações, "
@echo "use as seguintes opções de 'gcc':"
@echo " -I \$(INCDIR) -L \$(LIBDIR) -l\$(LIBNOME)\"."
@echo "*****"

limpa:
@echo "Limpendo sujeira ..."
@rm -f *% *.bak *~

faxina: limpa
@echo "Limpendo tudo ..."
@rm -rf *.o lib\$(LIBNOME).*