

# Alguns Tópicos da Linguagem C

parte II

# Struct

```
typedef struct {  
    double x;  
    double y;  
} complex;
```

```
struct complex{  
    double x;  
    double y;  
};
```

# Struct

```
typedef struct {  
    double x;  
    double y;  
} complex;
```

```
int main() {  
    complex var;  
    ...  
}
```

```
struct complex {  
    double x;  
    double y;  
};
```

```
int main() {  
    struct complex var;  
    ...  
}
```

# Struct

## Acesso Indireto

```
typedef struct {  
  
    double x;  
  
    double y;  
  
} complex;  
  
int main(){  
  
    complex* z1=(complex *)malloc(sizeof(complex));  
  
    double re = z1->x;  
  
    double im = z1->y;  
  
}
```

## Acesso Direto

```
typedef struct{  
  
    double x;  
  
    double y;  
  
} complex;  
  
int main(){  
  
    complex z1;  
  
    double re = z1.x;  
  
    double im = z1.y;  
  
}
```

# Passagem de Parâmetros

Por apontadores

```
complex* complex_add(complex* z, complex* w) {  
    return complex_new(z->x + w->x, z->y + w->y);  
}
```

# Passagem de Parâmetros

Por valor

```
complex complex_add(complex z, complex w) {  
    complex r;  
  
    r.x = z.x + w.x;  
  
    r.y = z.y + w.y;  
  
    return r;  
}
```

# Biblioteca estática

```
$ gcc -Wall -c complex_*.c
```

# Biblioteca estática

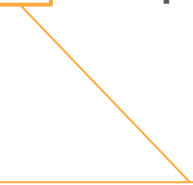
```
$ gcc -Wall -c complex_*.c
```

Ativa todas as  
mensagens de aviso  
(warning)



# Biblioteca estática

```
$ gcc -Wall -c complex_*.c
```



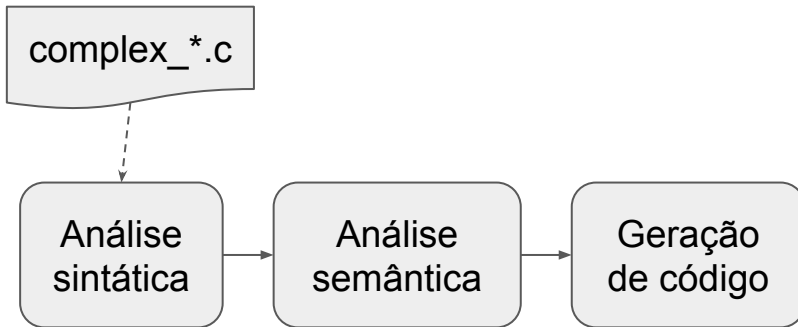
**Compilar os arquivos fonte** sem fazer o linking.  
Gera ficheiros binários ou ficheiros objeto (.o)

# Biblioteca estática

```
$ gcc -Wall -c complex_*.c
```

**Compilar os arquivos fonte** sem fazer o linking.  
Gera ficheiros binários ou ficheiros objeto (.o)

## Compilação



```
$ gcc -c  
complex_*.c
```

Para aqui e  
gera binário  
`complex_*.o`

# Biblioteca estática

```
$ ar -rc libcomplex.a complex_*.o
```

**archiver:** usado para  
criar, modificar e extrair  
bibliotecas estáticas.  
Resulta no ficheiro  
*nome\_biblioteca.a*

# Biblioteca estática

```
$ ar -rc libcomplex.a complex_*.o
```

Adicionar os  
ficheiros .o à  
biblioteca, se já  
existirem serão  
reescritos

# Biblioteca estática

\$ ar -rc libcomplex.a complex\_\*.o

Adicionar os  
ficheiros .o à  
biblioteca, se já  
existirem serão  
reescritos

Cria a  
biblioteca se  
ela ainda não  
existir

# Biblioteca estática

\$ ar -rclibcomplex.a complex\_\*.o

Adicionar os  
ficheiros .o à  
biblioteca, se já  
existirem serão  
reescritos

Cria a  
biblioteca se  
ela ainda não  
existir

Nome da  
biblioteca  
estática (.a)

# Biblioteca estática

\$ ar **-t** libcomplex.a

Exibe o conteúdo  
da biblioteca.  
Ficheiros objeto  
(.o)

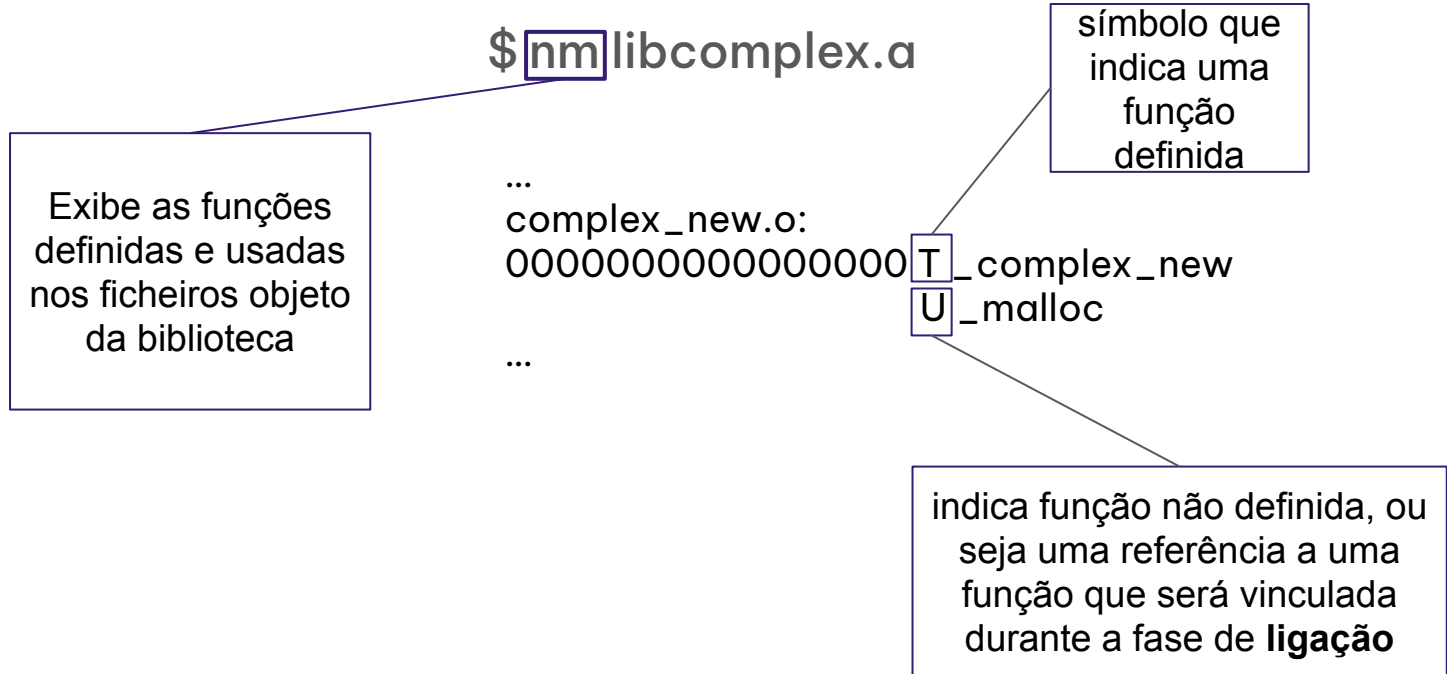
# Biblioteca estática

\$nm libcomplex.a

Exibe as funções  
definidas e usadas  
nos ficheiros objeto  
da biblioteca




# Biblioteca estática



# Biblioteca estática

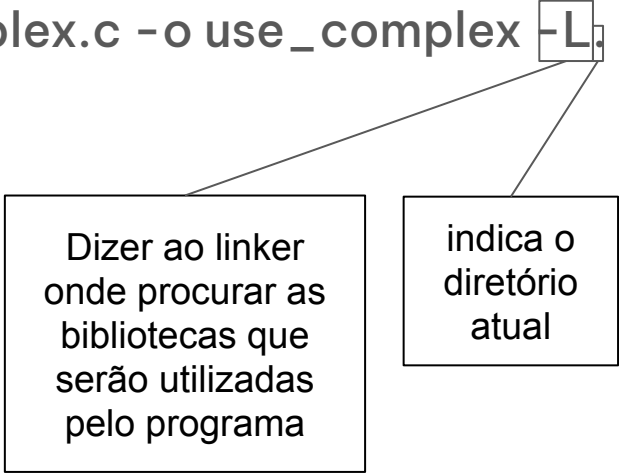
```
$ gcc -Wall use_complex.c -o use_complex -L. -lcomplex -lm
```



Dizer ao linker  
onde procurar as  
bibliotecas que  
serão utilizadas  
pelo programa

# Biblioteca estática

```
$ gcc -Wall use_complex.c -o use_complex -L. -lcomplex -lm
```



Dizer ao linker  
onde procurar as  
bibliotecas que  
serão utilizadas  
pelo programa

indica o  
diretório  
atual

# Biblioteca estática

```
$ gcc -Wall use_complex.c -o use_complex -L. -lcomplex -lm
```

Dizer ao linker onde procurar as bibliotecas que serão utilizadas pelo programa

Indica o diretório atual

O programa será vinculado a biblioteca estática **libcomplex.a**. O prefixo **lib** e a extensão **.a** são omitidos.

# Biblioteca Dinâmica

```
$ gcc -c -Wall fPIC complex_*.c
```

Esse flag gera código independente de posição (Position Independent Code). O código compilado com **-fPIC** pode ser carregado em qualquer endereço de memória, o que é essencial para criar bibliotecas compartilhadas.

Resultado: Ficheiros binários (.o) gerados a partir do código-fonte (.c), com a opção **-fPIC**, prontos para serem usados na criação de uma biblioteca dinâmica.

# Biblioteca Dinâmica

```
$ gcc -shared -o libcomplex.so complex_*.o
```

Esse flag indica a criação de uma biblioteca dinâmica. Uma biblioteca dinâmica é um ficheiro **.so** (Shared Object) no Linux, que pode ser carregada em tempo de execução por outros programas.

Especifica o nome da biblioteca. O prefixo padrão para bibliotecas é **lib** e a extensão **.so** é usada para bibliotecas dinâmicas no Unix.

# Biblioteca Dinâmica

\$ **nm** libcomplex.so

O comando mostra a tabela de símbolos de ficheiros binários, como executáveis e bibliotecas. A tabela de símbolos lista todas as funções, variáveis globais e outras entidades definidas ou referenciadas pelo código.

# Biblioteca Dinâmica

\$ **nm** libcomplex.so

O comando mostra a tabela de símbolos de ficheiros binários, como executáveis e bibliotecas. A tabela de símbolos lista todas as funções, variáveis globais e outras entidades definidas ou referenciadas pelo código.

**U**\_atan2  
0000000000000000c60 T\_complex\_add  
0000000000000000ca0 T\_complex\_arg  
0000000000000000cd0 **T**\_complex\_conj

Símbolo que indica uma função não definida (referenciada, mas não implementada neste arquivo/biblioteca)

Símbolo que indica uma função definida (no segmento do código)



# Biblioteca Estática x Biblioteca Dinâmica

- Biblioteca estática (.a): O código é incluído diretamente no executável durante o **processo de ligação**.
- Resulta em executáveis maiores.
- Biblioteca Dinâmica (.so): O código é carregado dinamicamente **em tempo de execução**.
- Executáveis que usam bibliotecas dinâmicas são menores e podem compartilhar a mesma biblioteca entre vários programas, o que pode resultar em uma otimização da memória.