



Universidade
Federal de
Sergipe

Universidade Federal de Sergipe
Departamento de Sistemas de Informação

SINF0064 – Programação II

Prof. Dr. Alcides Xavier Benicasa

alcides@ufs.br

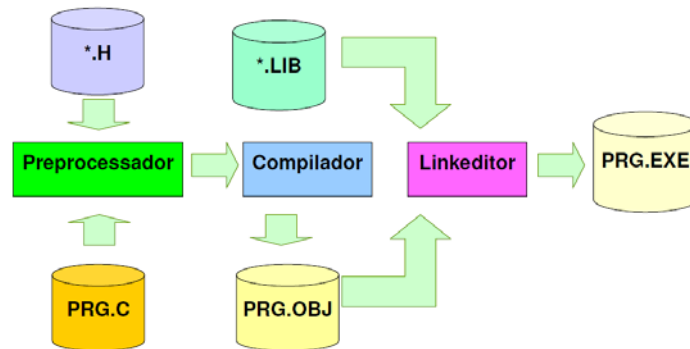
2

Roteiro

- Fases na Criação de um Executável
- Criação e chamada de Bibliotecas.
 - ◆ Compilação Separada
 - ◆ Arquivos de Implementação
- *Namespaces*
- Comandos de Controle de Vídeo

Fases na Criação de um Executável

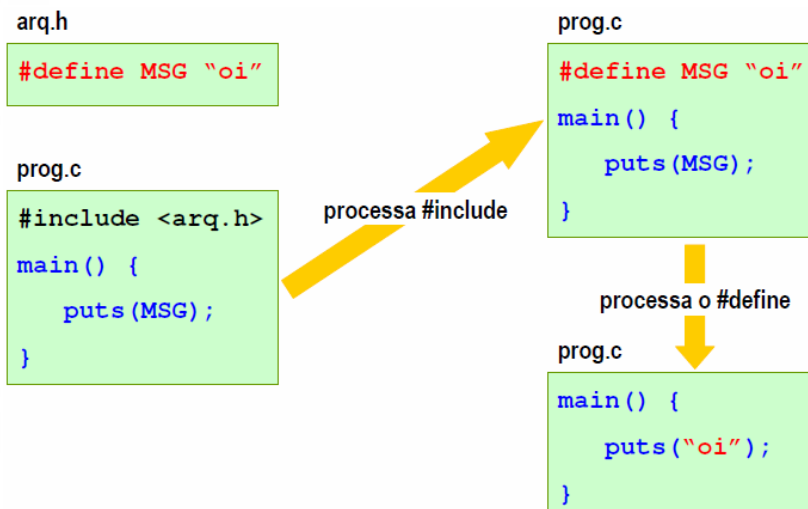
- Todo programa C, antes de chegar ao compilador é tratado por um módulo adicional denominado **pré-processador**.



3/37

UFS – Programação II – Prof. Dr. Alcides Xavier Benicasa

Pré-processamento



4/37

UFS – Programação II – Prof. Dr. Alcides Xavier Benicasa

Compilação e Linkedição

prog.c

```
main() {  
    puts("oi");  
}
```

compilação

Supondo
uma máquina
hipotética!

prog.obj

```
0100 push ds:0000  
0106 call cs:010A  
0109 ret
```

linkedição

prog.exe

```
0100 push ds:0000  
0106 call cs:0109  
0109 ret  
010A mov [sp], dx  
010E mov 30, ah  
0112 int 21  
0115 ret
```

5/37

UFS – Programação II – Prof. Dr. Alcides Xavier Benicasa

A diretiva #define

■ Substituição simples:

```
#define diga puts  
#define oi    "\nOlá, tudo bem?"  
main() {  
    diga(oi);  
}
```

```
main() {  
    puts("\nOlá, tudo bem?");  
}
```

6/37

UFS – Programação II – Prof. Dr. Alcides Xavier Benicasa

Exercício

- Inclua diretivas no programa, para que ele possa ser compilado.

```
#include <stdio.h>
programa
inicio
    diga("Olá!");
fim
```

7/37

UFS – Programação II – Prof. Dr. Alcides Xavier Benicasa

A diretiva #define

- Substituição parametrizada (**macro**):

```
#define quad(n) n*n

main() {
    printf("%d", quad(2) );
}
```

```
main() {
    printf("%d", 2*2 );
}
```

8/37

UFS – Programação II – Prof. Dr. Alcides Xavier Benicasa

Problemas com macros

```
#define quad(n) n*n  
main() {  
    printf("%d", quad(2+3) );  
}
```

```
main() {  
    printf("%d", 2+3*2+3 );  
}
```

Problemas com macros

```
#define quad(n) (n)*(n)  
main() {  
    printf("%d", 100/quad(2+3) );  
}
```

```
main() {  
    printf("%d", 100/(2+3)*(2+3) );  
}
```

Versão final da macro quad()

```
#define quad(n) ((n)*(n))
```

- REGRA: utilize sempre uma expressão completamente balanceada (**parentetizada**), ao definir uma macro!

Criação e chamada de Bibliotecas

A diretiva `#include`

- `#include` inclui no código-fonte uma cópia do conteúdo do arquivo indicado.
- Arquivos de inclusão padrão:
 - ◆ `#include <...>`
- Arquivos de inclusão do usuário:
 - ◆ `#include "..."`

Compilação Separada

- Partes do programa
 - ◆ Mantidas em **arquivos separados**
 - ◆ **Compilados** separadamente
 - ◆ **Ligados** antes do programa ser executado
- Se a **implementação mudar** → somente aquele arquivo precisa ser modificado
 - ◆ Programas '**Usuários**' não precisam ser modificados
- Separação '**Completa**'
 - ◆ Mudança da implementação → Não tem impacto sobre nenhum outro programa

Arquivos de Cabeçalho de Classe

■ Programas usuários deverão incluir:

- ◆ `#include "minhasdefinicoes.h"`
- ◆ As **aspas** indicam que você escreveu o cabeçalho
 - Encontra-se em '**sua**' pasta de trabalho
- ◆ Lembre-se das bibliotecas include; ex.: `<iostream>`
 - `< >` indicam arquivo de cabeçalho da **biblioteca predefinida**
 - E encontra-se na pasta "**library**"

15/37

UFS – Programação II – Prof. Dr. Alcides Xavier Benicasa

Exemplo de Biblioteca – Exemplo 01 C/C++

```
#include <locale.h>
#include <conio.h>
#include <iostream>
#include "dttime.h"
using namespace std;

int main( )
{
    setlocale(LC_ALL, "portuguese");
    cout << "Você pode digitar meia-noite como 0:00 ou 24:00,\n"
    << "mas eu sempre escreverei 0:00.\n"
    << "Digite a hora e minutos em notação de 24 horas: ";
    cin >> clock.hour >> clock.minute;

    oldClock = clock;
    advance(15);
    cout << "Você digitou " << oldClock.hour << ":" << oldClock.minute
    << endl;
    cout << "15 minutos depois a hora será "
    << clock.hour << ":" << clock.minute << endl;

    advance(2, 15);
    cout << "2 horas e 15 minutos depois disso\n"
    << "a hora será "
    << clock.hour << ":" << clock.minute << endl;

    getch(); return 0;
}
```

```
C:\Users\ALCIDE~1\WORKSP~3\NA02_E~1\bin\Debug\EX01_G~1.EXE
Você pode digitar meia-noite como 0:00 ou 24:00,
mas eu sempre escreverei 0:00.
Digite a hora e minutos em notação de 24 horas: 20
45
Você digitou 20:45
15 minutos depois a hora será 21:0
2 horas e 15 minutos depois disso
a hora será 23:15
```


Exemplo de Biblioteca – Exemplo 01 C/C++

```
struct DigitalTime {  
    int hour;  
    int minute;  
} clock, oldClock;  
  
void advance(int hoursAdded, int minutesAdded);  
  
void advance(int minutesAdded);  
  
void advance(int hoursAdded, int minutesAdded)  
{  
    clock.hour = (clock.hour + hoursAdded)%24;  
    advance(minutesAdded);  
}  
  
void advance(int minutesAdded)  
{  
    int grossMinutes = clock.minute + minutesAdded;  
    clock.minute = grossMinutes%60;  
  
    int hourAdjustment = grossMinutes/60;  
    clock.hour = (clock.hour + hourAdjustment)%24;  
}
```

Namespaces

Namespaces

- Um **namespace** é um mecanismo para expressar um agrupamento lógico

- Sintaxe:

```
namespace nomeDoNamespace {  
    corpoDoNamespace  
}
```

19/37

UFS – Programação II – Prof. Dr. Alcides Xavier Benicasa

Namespaces

- Podemos utilizar um **namespace** para agrupar diversas funções, classes, variáveis
- Por exemplo, se tivermos muitas funções para realização de operações matemáticas podemos criar um **namespace** para todas

20/37

UFS – Programação II – Prof. Dr. Alcides Xavier Benicasa

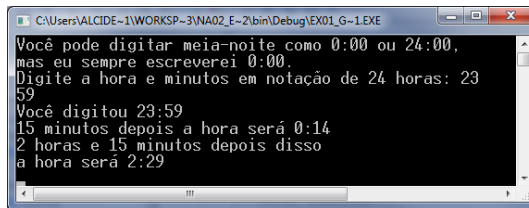
Exemplo de Biblioteca – Exemplo 02 C/C++

```
#include <locale.h>
#include <conio.h>
#include <iostream>
#include "dttime.h"
using namespace std;
using namespace dttime;
```

```
int main( )
{
    setlocale(LC_ALL, "portuguese");
    cout << "Você pode digitar meia-noite como 0:00 ou 24:00,\n"
          << "mas eu sempre escreverei 0:00.\n"
          << "Digite a hora e minutos em notação de 24 horas: ";
    cin >> clock.hour >> clock.minute;

    oldClock = clock;
    advance(15);
    cout << "Você digitou " << oldClock.hour << ":" << oldClock.minute
          << endl;
    cout << "15 minutos depois a hora será "
          << clock.hour << ":" << clock.minute << endl;

    advance(2, 15);
    cout << "2 horas e 15 minutos depois disso\n"
          << "a hora será "
          << clock.hour << ":" << clock.minute << endl;
    getch(); return 0;
}
```



Exemplo de Biblioteca – Exemplo 02 C/C++

```
namespace dttime {
    struct DigitalTime {
        int hour;
        int minute;
    } clock, oldClock;

    void advance(int hoursAdded, int minutesAdded);

    void advance(int minutesAdded);

    void advance(int hoursAdded, int minutesAdded) {
        clock.hour = (clock.hour + hoursAdded)%24;
        advance(minutesAdded);
    }

    void advance(int minutesAdded)
    {
        int grossMinutes = clock.minute + minutesAdded;
        clock.minute = grossMinutes%60;

        int hourAdjustment = grossMinutes/60;
        clock.hour = (clock.hour + hourAdjustment)%24;
    }
}
```

Namespaces

- Podemos acessar os membros de um **namespace** de duas maneiras diferentes:
 - ◆ Usando o operador **::**
 - ◆ Através do comando **using namespace**

23/37

UFS – Programação II – Prof. Dr. Alcides Xavier Benicasa

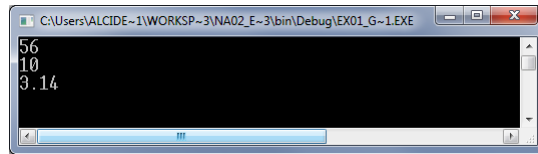
Exemplo de Namespaces – Exemplo 03 C/C++

```
namespace Mat {  
    const float PI = 3.14;  
  
    int maximo(int a, int b) {  
        return (a > b) ? a : b;  
    }  
  
    int minimo(int a, int b) {  
        return (a < b) ? a : b;  
    }  
}
```

Exemplo de Namespaces – Exemplo 03 C/C++ (cont.)

```
#include <conio.h>
#include <iostream>
#include "exemploNamespace.h"

int main()
{
    std::cout << Mat::maximo(10, 56) << std::endl;
    std::cout << Mat::minimo(10, 56) << std::endl;
    std::cout << Mat::PI << std::endl;
    getch();
    return 0;
}
```



Namespaces

- **using namespace** podem ser utilizados quando tivermos mais de uma função com o mesmo protótipo.
- Exemplo:
 - ◆ *exemploNamespaceFuncoes.cpp*

Exemplo de Namespaces – Exemplo 04 C/C++

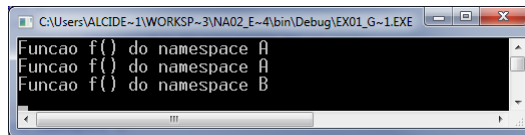
```
#include <conio.h>
#include <iostream>
using namespace std;

namespace A {
    void f() {
        cout << "Funcao f() do namespace A" << endl;
    }
}

namespace B {
    void f() {
        cout << "Funcao f() do namespace B" << endl;
    }
}

void g() {
    using namespace A;
    f();
}

int main() {
    A::f();
    g();
    B::f();
    getch();
    return 0;
}
```



Comandos de Controle de Vídeo

```
#include <conio2.h>
```

Configurando o CodeBlocks

■ Instruções para usar o **conio2** no **CodeBlocks**:

1. Copiar o arquivo **conio2.h** para **C:\Arquivos de Programas(x86)\CodeBlocks\MingW\include**
2. Copiar o arquivo **libconio.a** para **C:\Arquivos de Programas(x86)\CodeBlocks\MingW\lib**
3. Abrir o **CodeBlocks** e entrar no menu **Settings** → **Compiler**
4. Clicar no botão "**Linker Settings**"
5. Abaixo do quadro "**Link libraries**", clicar no botão "**Add**"
6. Escrever na janelinha "**Add library**" o seguinte: **libconio.a**
7. No programa, colocar: **#include <conio2.h>**

29/37

UFS – Programação II – Prof. Dr. Alcides Xavier Benicasa

Configurando o CodeBlocks

■ Algumas funções do **conio2**:

- ◆ **textbackground()**
 - Seta a cor de fundo
- ◆ **textcolor()**
 - Seta a cor do texto
- ◆ **clrscr()**
 - Limpa a tela
- ◆ **gotoxy()**
 - Coloca o cursor em uma posição da tela

30/37

UFS – Programação II – Prof. Dr. Alcides Xavier Benicasa

Exemplo de posicionamento do cursor - Exemplo 05 C/C++

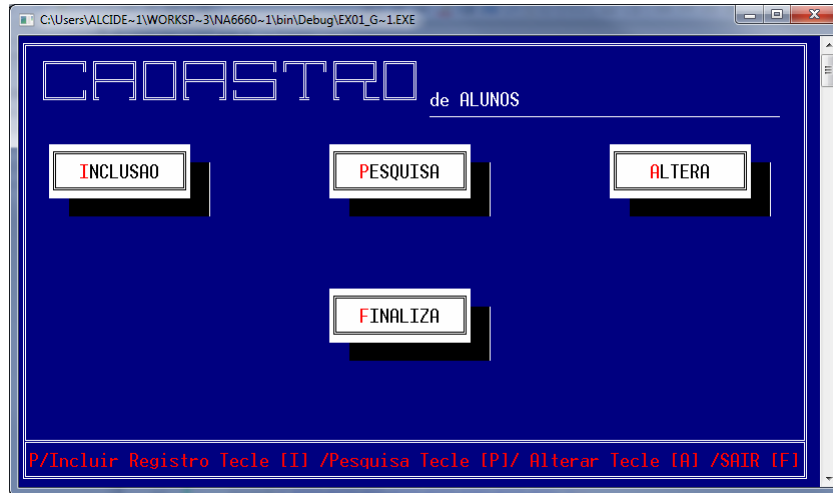
```
#include <conio.h>
#include <conio2.h>
#include <stdio.h>

int main() {
    int col, lin;
    char c;
    printf("Pressione uma tecla...");
    getch();
    clrscr();
    printf ("Posicionar cursor? (s/n): ");
    c = getch();
    while (c == 's' || c == 'S') {
        clrscr();
        gotoxy(1, 1); printf("Digite a coluna: ");
        gotoxy(1, 2); printf("Digite a linha : ");
        gotoxy(18,1); scanf("%d", &col);
        gotoxy(18,2); scanf("%d", &lin);
        gotoxy(col, lin);
        getch();
        clrscr();
        gotoxy(1,1);
        printf("Posicionar cursor? (s/n): ");
        c = getch();
    }
    return(0);
}
```

Exemplo do Formato das Próximas Aplicações

utilizando conceitos vistos...

Exemplo de posicionamento do cursor - Exemplo 06 C/C++



Exemplo de posicionamento do cursor - Exemplo 06 C/C++

```
#include <windows.h> #include <stdio.h> #include <conio2.h> #include <ctype.h>
#include "cad.h" #include "tela.h" using namespace cad;
int main () {
    int i;
    char opcao ;
    cadastro alunos[3];
    desenhaTela();
    opcao = getch();
    while ((opcao=='I') || (opcao=='i') || (opcao=='P') || (opcao=='p') || (opcao=='A') ||
           (opcao=='a')) {
        switch (toupper(opcao)) {
            case 'I' : for (i=1; i<=3; i++) {
                        telaIncluir(i);
                        incluir(alunos,i); } break;
            case 'P' : for (i=1; i<=3; i++) {
                        telaPesquisar(i);
                        pesquisar(alunos,i); } break;
            case 'A' : for (i=1; i<=3; i++) {
                        telaAlterar(i);
                        alterar(alunos,i); } break;
        }
        desenhaTela();
        opcao = getch();
    }
    telaFinaliza();
    clrscr();
    return 0;
}
```

Exemplo de posicionamento do cursor - Exemplo 06 C/C++

■ Demais Códigos

◆ tela.h e

◆ cad.h

Tabela ASCII				Extended ASCII characters											
DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC
128	80h	Ç	160	A0h	á	192	C0h	Ł	224	E0h	Ó	256	F0h	à	288
129	81h	ü	161	A1h	í	193	C1h	ł	225	E1h	ô	257	F1h	á	289
130	82h	é	162	A2h	ó	194	C2h	Ł	226	E2h	ö	258	F2h	â	290
131	83h	â	163	A3h	ú	195	C3h	ł	227	E3h	õ	259	F3h	ã	291
132	84h	ä	164	A4h	ñ	196	C4h	Ł	228	E4h	ö	260	F4h	ä	292
133	85h	å	165	A5h	Ñ	197	C5h	ł	229	E5h	ù	261	F5h	ä	293
134	86h	à	166	A6h	ª	198	C6h	Ł	230	E6h	û	262	F6h	å	294
135	87h	ç	167	A7h	º	199	C7h	ł	231	E7h	ü	263	F7h	æ	295
136	88h	ê	168	A8h	¿	200	C8h	Ł	232	E8h	ý	264	F8h	ç	296
137	89h	ë	169	A9h	®	201	C9h	ł	233	E9h	ÿ	265	F9h	è	297
138	8Ah	è	170	AAh	™	202	CAh	Ł	234	EAh	ÿ	266	FAh	é	298
139	8Bh	í	171	ABh	½	203	CBh	ł	235	EBh	ÿ	267	FBh	ê	299
140	8Ch	ï	172	ACH	¼	204	CDh	Ł	236	ECh	ÿ	268	FBh	ë	300
141	8Dh	ï	173	ADh	¾	205	CDh	ł	237	EDh	ÿ	269	FDh	ì	301
142	8Eh	Ā	174	Aeh	«	206	CEh	Ł	238	Eeh	ÿ	270	FDh	í	302
143	8Fh	Ā	175	Afh	»	207	CFh	ł	239	Efh	ÿ	271	FDh	î	303
144	90h	Ē	176	B0h	⌘	208	D0h	Ł	240	F0h	ÿ	272	FDh	ï	304
145	91h	æ	177	B1h	⌘	209	D1h	ł	241	F1h	ÿ	273	FDh	ï	305
146	92h	Æ	178	B2h	⌘	210	D2h	Ł	242	F2h	ÿ	274	FDh	ï	306
147	93h	ø	179	B3h	⌘	211	D3h	ł	243	F3h	ÿ	275	FDh	ï	307
148	94h	ø	180	B4h	⌘	212	D4h	Ł	244	F4h	ÿ	276	FDh	ï	308
149	95h	ø	181	B5h	⌘	213	D5h	ł	245	F5h	ÿ	277	FDh	ï	309
150	96h	ù	182	B6h	⌘	214	D6h	Ł	246	F6h	ÿ	278	FDh	ï	310
151	97h	ù	183	B7h	⌘	215	D7h	ł	247	F7h	ÿ	279	FDh	ï	311
152	98h	ÿ	184	B8h	⌘	216	D8h	Ł	248	F8h	ÿ	280	FDh	ï	312
153	99h	ÿ	185	B9h	⌘	217	D9h	ł	249	F9h	ÿ	281	FDh	ï	313
154	9Ah	ÿ	186	BAh	⌘	218	DAh	Ł	250	FAh	ÿ	282	FDh	ï	314
155	9Bh	ø	187	BBh	⌘	219	DBh	ł	251	FBh	ÿ	283	FDh	ï	315
156	9Ch	£	188	BCh	⌘	220	DCh	Ł	252	FCh	ÿ	284	FDh	ï	316
157	9Dh	Ø	189	BDh	⌘	221	DDh	ł	253	FDh	ÿ	285	FDh	ï	317
158	9Eh	×	190	BEh	⌘	222	DEh	Ł	254	FEh	ÿ	286	FDh	ï	318
159	9Fh	f	191	BFh	⌘	223	DFh	ł	255	FFh	ÿ	287	FDh	ï	319

■ Exemplo

◆ "=" //ALT+205

◆ "ŕ" //ALT+201

◆ "ŕ" //ALT+188

◆ "ŕ" //ALT+187

◆ "ŕ" //ALT+186

◆ "ŕ" //ALT+200

Referências



- SAVITCH, Walter J. *C++ Absoluto*. Tradução Claudia Martins. 1.ed - São Paulo: Addison Wesley, 2004.
- Stroustrup, Bjarne. *The C++ Programming Language, Special Edition*.
- Eckel, Bruce. *Thinking in C++, 2nd ed. Volume 1*.
- Agradecimentos:
 - ◆ Notas de Aula de Gustavo (ghcp@cin.ufpe.br) do curso de C/C++
 - ◆ Notas de Aula de Luiz Felipe Sotero do curso de C/C++