# Programação I
# PRG29002

Engenharia de Telecomunicações 2ª Fase

Professor: Cleber Jorge Amaral

2016-1

INSTITUTO FEDERAL
SANTA CATARINA

# Introdução ao C (continuação)

- C é uma linguagem “compilada”, ou seja, de um código fonte (escrito em C) são gerados códigos de máquina formando um ou mais arquivos executáveis e inteligíveis apenas para o computador
- Há diversos compiladores e estes podem ter algumas diferenças de comportamento e aceitarem diferentes parametrizações
- Um código é compilado para um sistema operacional específico e uma arquitetura de processador, portanto, um código compilado para um S.O. não tem qualquer garantia de funcionamento em outros sistemas. Da mesma forma um código que roda em um PC, não tem qualquer garantia de rodar em outras arquiteturas diversas
- Em oposição ao código compilado temos o código interpretado

# Introdução ao C (continuação)

- Sempre que um código fonte é modificado se faz necessário nova compilação para que as modificações façam efeito na execução
- As variáveis que serão utilizadas pelo programa devem ser listadas antecipadamente
- A linguagem C tem um conjunto de palavras reservadas, que não podem ser utilizadas para outro propósito se não o que está definido na estrutura da linguagem
  - Exemplos: break, case, if, for, while, return,...

# Introdução ao C (continuação)

- O C permite que trabalhemos com bibliotecas (lib) que são conjuntos de funções que realizam certas tarefas
- Além de podermos criar nossas próprias bibliotecas com funções úteis que podemos reutilizar em vários programas, também podemos nos apropriar de diversas libs já desenvolvidas, sejam padrão ANSI (libc) ou não, desta forma não precisamos “reinventar a roda” e já sair de largada com várias funcionalidades
  - Exemplos: <stdio.h>, <math.h>, <complex.h>, <float.h>, <string.h>, etc. (são 24 padrão ANSI no total)

# Passo a passo da compilação de um projeto em C

1) Edição: atividade feita pelo programador
2) Preprocessamento: compilador processa o código e ignorando comentários, fazendo associações de constantes e controle de código através de diretivas especiais de compilação
3) Compilação: criação do código-objeto, é a tradução da linguagem C em linguagem de máquina
4) Linkagem: associação de diferentes código-objeto e bibliotecas
5) Carregamento: carrega o programa em memória
6) Execução: cpu realiza a execução das instruções passo a passo, armazenando os resultados em memórias definidas pelo programa e pilhas de dados para controle

# Comentários

- Como vimos podemos incluir no programa fonte textos livres que ajudam na compreensão do código
- Os comentários são ignorados pelo compilador, não se tornam código de máquina
- Para incluir comentários inicie com /* digitando então o comentário aqui e terminando com */
  - Este formato permite que digitemos varias linhas de comentários, normalmente é utilizado para textos mais extensos
- A maioria dos compiladores também aceita o formado //comentário, que serve para incluir um comentário de apenas uma linha, apenas os caracteres depois do // serão ignorados e neste caso o terminador é o sinal de nova linha que normalmente está oculto

# Operadores aritméticos

- Operadores aritméticos
  - “+” adição
  - “-” subtração
  - “*” multiplicação
  - “/” divisão
  - “%” resto da divisão
- Por padrão, multiplicações e divisões são operadas antes de somas e subtrações

# Uso de parênteses

- Devemos utilizar parênteses para agrupar operações e definir a sequencia mais adequada. O compilador vai sempre resolver o que está dentro dos parênteses primeiro, de “dentro para fora” quando houver mais de um nível
- Exemplos
  - 1+2*3 = 7 é o mesmo que 1+(2*3)
  - (1+2)*3 = 9
  - 1+2*3+4*5 = 27 é o mesmo que 1+(2*3)+(4*5)
  - (((1+2)*3)+4)*5 = 65

# Escrevendo mensagens na tela

- A função printf da lib stdio é bastante completa para esta tarefa, permite escrever mensagens com múltiplos argumentos.
- Formato printf (“string de controle”, lista de argumentos);
- Exemplo:
  - printf(“Olá Mundo!\n”);
  - printf(“Digite sua idade:\n”);
  - printf(“Sua idade é: %d”,idade);

Algumas strings de controle
%c caractere
%d decimal
%e notação científica
%f ponto flutuante
%o formato octal
%x hexadecimal
%s cadeia de caracteres
%lf double

# Lendo o teclado do usuário

- A função scanf da lib stdio é bastante útil para esta tarefa, ela aguarda que o usuário entre com uma informação e tecle [ENTER] no final.
- Está função é blocante, ou seja, o programa fica parado esperando a entrada de dados para então dar continuidade a execução
- Formato scanf (“string de controle”, lista de argumentos);
- Exemplo:
  - scanf(“%d”,&idade);

# Exercícios (lista 4)

- Implemente um programa em C que calcula a média de dois números reais digitados pelo usuário e imprime em tela a resposta deste cálculo
- Implemente um programa em C que recebe “a”, “b” e “c”, calcula e exibe o delta (delta = 2b-4ac)
- Implemente um programa em C que recebe duas datas fornecidas pelo usuário (três números inteiros cada: dia, mês e ano com 4 dígitos). Deve ser calculada qual a maior data e exibi-la em tela
- Implemente um programa em C que calcule a Potência dissipada em um resistor e a corrente dados valores de V e R. O mesmo programa deve ser capaz de calcular a R dados P e V e R dados V e I. Faça então um menu inicial para que o usuário possa selecionar a opção de cálculo desejada (pesquise sobre if...else para resolver este problema)

# Obrigado pela atenção e participação!

Cleber Jorge Amaral (cleber.amaral@ifsc.edu.br)

Horários de atendimento (2016-1):
Quintas-feiras as 17:30 no laboratório de Programação

Sextas-feiras as 17:30 no Laboratório de Meios de Transmissão