Curso de C/C++ Avançado

Aula 4 – Biblioteca Padrão do C ANSI

Allan Lima – http://allanlima.wordpress.com

creative commons
COMMONS DEED

SOME RIGHTS RESERVED

- *Você pode:*
  - copiar, distribuir, exibir e executar a obra
  - criar obras derivadas
  - fazer uso comercial da obra
- *Sob as seguintes condições:*
  - **Atribuição**. Você deve dar crédito ao autor original, da forma especificada pelo autor ou licenciante.
  - **Compartilhamento pela mesma Licença**. Se você alterar, transformar, ou criar outra obra com base nesta, você somente poderá distribuir a obra resultante sob uma licença idêntica a esta.
  - Para cada novo uso ou distribuição, você deve deixar claro para outros os termos da licença desta obra.
  - Qualquer uma destas condições podem ser renunciadas, desde que Você obtenha permissão do autor.
- *Veja aqui a licença completa*

cit

A Biblioteca de C ANSI

- *Possui um conjunto de funções e tipos de dados que podemos utilizar nos nossos programas*
- *As funções são portáveis e possuem boa performance*
- *Pode ser muito útil para nos poupar tempo*
- *Não reinvente a roda!*

Biblioteca de C ANSI

- *Podemos utilizar a biblioteca através da diretiva #include*

```
#include <stdio.h>
#include <stdlib.h>
#inlcude <string.h>
...
```

Principais Headers

Header	Descrição
stdio.h	Funções de E/S padrão
stdlib.h	Funções diversas
string.h	Funções para manipulação de strings
math.h	Funções matemáticas
time.h	Manipulação de tempo/datas
stdargs.h	Criação de funções como scanf e printf
limits.h	Macros com os limites para tipos inteiros
float.h	Limites para pontos flutuantes

stdio.h

- *Contém diversas funções para o tratamento de entrada e saída*
- *Contém as principais funções que utilizamos:*
  - scanf(),printf(), fopen(), fclose(), ...

string.h

- *Contém um conjunto de funções para manipulação de strings*
- *Principais funções:*
  - strcmp(), strlen(), strcpy(), strcat(), ...
- *Já foi amplamente abordado na aula sobre strings*

stdlib.h

- *Possui diversas funções de utilidade geral*
- *Útil para*
  - Alocação dinâmica de memória
  - Conversão de tipos
  - Controle de processos
  - Ordenação
  - Busca
  - Geração de números pseudo-aleatórios
  - ...

stdlib.h

- *Funções para conversão de tipos:*
  - *int atoi(char *str);*
    - Converte uma string para um int
  - *float atof(char *str);*
    - Converte uma string para um float
- *Exemplo:*
  - *exemploConversao.c*

stdlib.h

- *Funções para controle de processos:*
  - *void exit(int status);*
    - Encerra o programa
  - *int system(const char *comando);*
    - Executa um comando do Sistema Operacional
    - Retorna 0 se o comando foi executado com sucesso, ou –1 caso contrário
  - *int atexit(void (* funcao) (void));*
    - Faz com que o programa chame a função passada como parâmetro no seu fim
    - Várias funções podem ser adicionadas e são chamadas na ordem inversa

Exemplo

- *exemploProcessos.c*

stdlib.h

- *Ordenação:*
  - void qsort(
    void *elementos,
    int numeroDeElementos,
    int tamanhoDoTipo,
    int (* funcDeComparacao) (
    const void *e1, const void *e2
)
    );

stdlib.h

- *Ordenação:*
  - A função qsort() é uma implementação genérica do algoritmo **Quick Sort**
  - ***elementos*** é o array que será ordenado
  - ***numeroDeElementos*** é quantidade de posições preenchidas no array
  - ***tamanhoDoTipo*** é tamanho em bytes do tipo dos elementos do array
  - ***funcDeComparacao*** é função que será utilizada para a comparação entre elementos do array

Exemplo

- *exemploQsort.c*

stdlib.h

- *Busca binária:*
  - void *bsearch(
    void *chave,
    void *elementos,
    int numeroDeElementos,
    int tamanhoDoTipo,
    int (* funcDeComparacao) (
    const void *e1, const void *e2
)
    );

stdlib.h

- ***chave*** *é um ponteiro para o elemento que se busca*
- ***elementos*** *é o array onde a busca será realizada*
- ***numeroDeElementos*** *é quantidade de posições preenchidas no array*
- ***tamanhoDoTipo*** *é tamanho em bytes do tipo dos elementos do array*
- ***funcDeComparacao*** *é função que será utilizada para a comparação entre elementos do array*
- *Retorna um ponteiro para o elemento encontrado ou NULL se a chave não existe*

stdlib.h

- *Ordenação:*
  - A função bsearch() é uma implementação genérica do algoritmo de **Busca Binária**
  - **Para que o algoritmo funcione o array deve esta ordenado de acordo com a função de comparação**

stdlib.h

- *Geração de números pseudo-aleatórios:*
  - *void srand(unsigned int semente);*
    - Inicializa o gerador de números pseudo-aleatórios
  - *int rand();*
    - Gera um número pseudo-aleatório entre 0 RAND_MAX
    - RAND_MAX é uma constante definida em stdlib.h e o seu valor depende do compilador e do S.O.

Exemplo

- *exemploRand.c*

math.h

- *Biblioteca padrão para operações matemáticas*
- *Principais funções:*
  - *int abs(int n);*
    - Retorna o valor absoluto (módulo) de n
  - *double pow(double x, double y);*
    - Retorna x ^ y (x elevando a y)
  - *double sqrt(double x)*
    - Retorna a raiz quadrada x

math.h

- *Principais funções (cont.):*
  - *double sin(double x);*
    - Calcula o seno de x
  - *double cos(double x);*
    - Calcula o co-seno de x
  - *double tan(double x);*
    - Calcula a tangente de x

Nestas funções x é expresso em radianos
PI radianos = 180 graus

Exemplo

- *exemploMath.c*

limits.h

- *Possui diversas macros com os limites para os tipos inteiros de C*
- *Principais macros:*

Constante	Significado
INT_MAX	Valor máximo do tipo int
INT_MIN	Valor mínimo do tipo int
LONG_MAX	Valor máximo do tipo long
LONG_MIN	Valor mínimo do tipo long
SHRT_MAX	Valor máximo do tipo short
SHRT_MIN	Valor mínimo do tipo short
CHAR_MAX	Valor máximo do tipo char
CHAR_MIN	Valor mínimo do tipo char

Exemplo

- *exemploLimites.c*

float.h

- *Possui diversas macros com os limites para os tipos de ponto flutuante de C*
- *Principais constantes:*

Constante	Descrição
DBL_MAX	Valor máximo do tipo double
DBL_MIN	Valor positivo mínimo do tipo double
DBL_DIG	Casas decimais do double
FLT_MAX	Valor máximo do tipo float
FLT_MIN	Valor positivo mínimo do tipo float
FLT_DIG	Casas decimais do tipo float

Exemplo

- *exemploFloat.c*

time.h

- *Contém diversas funções e estruturas definidas para a manipulação de tempo e data*
- *Principais tipos de dados:*

time_t	Representa o tempo em segundos (geralmente long int)
clock_t	Representa a quantidade de ciclos do processador (long int)
struct tm	Representa o tempo em forma de calendário com dia, mês, ano, ...

time.h

```
struct tm {
    int tm_sec; // segundos de 0 a 59
    int tm_min; // minutos de 0 a 59
    int tm_hour; // hora do dia de 0 a 23
    int tm_mday; // dia do mês de 1 em diante
    int tm_mon; // mês do ano de 0 a 11
    int tm_year; // ano desde 1900
    int tm_wday; // dia da semana de 0 a 6
    int tm_yday; // dia do ano
    int tm_isdst; // Daylight Saving Time flag
};
```

time.h

- *Principais funções:*
  - *clock_t clock();*
    - Retorna o número de ciclos do processador desde o inicio do programa
  - *time_t time(time_t *tod);*
    - Retorna o tempo corrente no calendário local, em caso de erro retorna -1. Se *tod* for diferente de NULL o tempo será armazenado em *tod*
  - *char *asctime(const struct tm *tptr);*
    - Converte *tptr* em uma string de 26 caracteres na representação de datas inglesa.

time.h

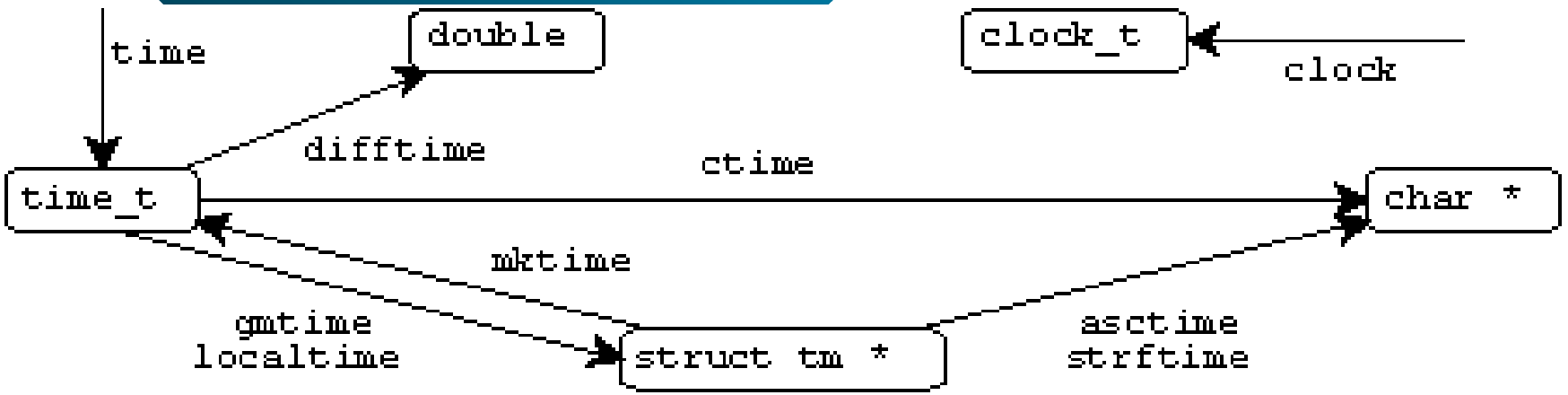
- *Principais funções (cont.):*
  - *char *ctime(const time_t *tod);*
    - Converte *tod* em uma string de 26 caracteres na representação de datas inglesa.
  - *double difftime(time_t t1, time_t t2);*
    - Retorna t1 – t2 em segundos
  - *struct tm *gmtime(const time_t *tod);*
    - Converte *tod* em uma struct tm no horário de Greenwich (GMT)

time.h

- *Principais funções:*
  - *struct tm *localtime(const time_t *tod);*
    - Converte *tod* em uma struct tm no horário local
  - *time_t mktime(struct tm *tptr);*
    - Converte *tptr* em um time_t

time.h

- *Resumo*



http://www.ccs.ucsd.edu/c/

Exemplo:

- *exemploTime.c*

stdarg.h

- *Header utilizado para implementação de funções como printf() e scanf()*
- *Possui macros para que possamos acessar todos os parâmetros*
- *Os parâmetros são tratados como uma lista*

stdarg.h

- *Exemplos de declarações de funções com número de parâmetros indeterminados:*
  - void f(int a, ...) { }
  - int g(char *a, ...) { }
- *Não podemos determinar o tipo dos parâmetros*

stdarg.h

- *O tipo va_list:*
  - Representa uma lista de variáveis
  - É utilizado para termos acesso aos parâmetros da função
  - No corpo da função devemos inicializa-lá e finaliza-lá

stdarg.h

- *Principais macros:*
  - *va_start(ap,v)*
    - Inicializa uma lista de parâmetros
    - *ap* é a lista a ser inicializada
    - *v* é um ponteiro para o primeiro parâmetro da lista
  - *va_arg(ap,t)*
    - Retorna o próximo parâmetro da lista
    - *ap* é lista de onde o parâmetro será lido
    - *t* é tipo do parâmetro
  - *va_end(ap)*
    - Encerra *ap*

Exemplo

- *exemploStdarg.c*

Exercícios

1) *Implemente um programa que gera um número pseudo-aleatório entre 0 e 10, em seguida, lê inteiros da entrada padrão até que o último número lido seja igual ao número gerado*

Exercícios

2)Crie um programa que lê 5 strings da entrada padrão, guarda elas em um array, ordena o array utilizando a função qsort() em seguida, fica lendo mais strings e verificando se elas pertencem ao array com a função bsearch() até a string “sair” seja digitada.

Exercícios

3) *Implemente uma função que soma dois inteiros e imprime o resultado na saída padrão, antes da soma a função deve verificar se esta não irá ultrapassar o valor máximo do inteiro. Se este erro for detectado um mensagem de alerta deve se impressa saída padrão*

Exercício

4)Escreva um programa que funciona como uma calculadora, com as funções do header math.h. A operação a ser utilizada e os parâmetros são passados como argumentos para a função main.

Parâmetro	Significado
a	Valor absoluto
p	Potência
s	Raiz quadrada
n	Seno
c	Co-seno
t	Tangente

Exercícios

- *Exemplos:*
  - meuPrograma a -10 // imprime 10
  - meuPrograma p 2 9 // imprime 512
  - meuPrograma c 0 // imprime 1

Exercícios

5) *Crie um programa que calcula o número de dias que faltam para uma data. O usuário deve entrar com a data e o programa imprime o número de dias que faltam. Lembre-se de setar os atributos não lidos da struct tm para 0.*

Exercícios

6) *Implemente uma função que calcula o máximo entre um número indefinido de inteiros*

Referências

- *Curso de C da UFMG*
  - http://ead1.eee.ufmg.br/cursos/C/
- *Standart C*
  - http://www.ccs.ucsd.edu/c/
- *cplusplus resources*
  - www.cplusplus.com
- *Slides de Gustavo (ghpc@cin.ufpe.br) do Curso C/C++*