PIP/CA - *Programa Interdisciplinar de Pós-Graduação em Computação Aplicada da UNISINOS*

- **Disciplina de Nivelamento - 2000/1:**

ALGORITMOS
&
ESTRUTURAS DE DADOS

- **Professor Responsável:**

Prof. Fernando Santos Osório
E-Mail: osorio@exatas.unisinos.br
Web: http://www.inf.unisinos.br/~osorio/

Ponteiros na Linguagem "C" :

- Em vez de usarmos variáveis, nomes que "escondem" os endereços de memória onde estão armazenados os dados, podemos acessar diretamente os dados através de seus endereços...

Declaração: **<*tipo_variável*> *<*nome_variável*>;**
Uso:
& <*nome_variável*> => Obtém o endereço da variável
*** <*nome_variável*> => Obtém o conteúdo apontado pelo ponteiro**
Operadores: ++,-- => Podemos realizar operações aritméticas...

ATENÇÃO: Declarar um ponteiro não implica em alocar a memória p/o dado!

Exemplo:


```
int a=10;          /* a == 10   */
int *ptr_int;      /* ponteiro */

ptr_int  = &a;     /* ptr_int => a */
*ptr_int = 5;      /* a == 5 !     */
```

```
int  vetor[10];

&(vetor[0]) == vetor
&(vetor[1]) == vetor+1
&(vetor[2]) == vetor+2 ...
```

Ponteiros na Linguagem "C" :

- Quando usamos a rotina *scanf* já estamos usando ponteiros...

```
scanf ("%d", &var_int);     /* Endereço de var_int */
scanf (%s", var_string);    /* Vetor de char */
```

- Alocação de memória:

calloc - Aloca memória, zerando os dados
malloc - Aloca memória, sem inicializar os dados
free - Libera um bloco de memória alocada previamente

void *calloc (<*quantidade_elementos*>, <*tamanho_elemento*>)

Exemplo: tabela_inteiros = calloc(10, sizeof (int));

void *malloc(<*quantidade_elementos*>, <*tamanho_elemento*>)

Exemplo: tabela_inteiros = malloc(10, sizeof (int));

void free (void *ponteiro)
Exemplo: free (tabela_inteiros);

Sub-Rotinas na Linguagem "C" : Procedures e Funções

- Todo bloco pode receber um nome, com parâmetros de E/S :

```
<tipo_retornado> nome_da_rotina ( <nome_vars_parâmetros> )
<declaração_dos_parâmetros>;
{
   <variáveis_locais>;
   <comandos>;
   return ( <resultado> );
}
```

- Exemplos:

```
main ( )
{
   printf("Hello world!\n");
}
```

```
void main ( argc, argv )
int argc;
char *argv [ ];
{
      printf("Nro. args: %d\n",argc);
}
```

Sub-Rotinas na Linguagem "C" : Procedures e Funções

- Exemplos de sub-rotinas : ***função - retorna um valor***

```
long fatorial ( num )                    /* Rotina Fatorial   - Função  */
int num;                                 /* Entrada:  num     - inteiro */
{                                        /* Saída     : fatorial - long */
    int aux;
    long resultado;                      /* Variáveis locais */

    resultado = 1;
    for (aux = num; num != 1; num--)
        resultado = resultado * aux;     /* Ou...     resultado *= aux; */
   return ( resultado );                 /* Retorna o resultado         */
}

main ( )
{
    long fat;
    printf ("Fatorial de 5 é : %ld \n", fatorial(5));
    fat = fatorial ( 6 );                /* Chama a função fatorial     */
}
```

Sub-Rotinas na Linguagem "C" : Procedures e Funções

- Exemplos de sub-rotinas : ***procedure - não retorna nada***

```
void exibe_media ( v1, v2 )              /* Rotina   : exibe_media      */
int v1, v2;                              /* Entrada:  v1, v2 - inteiros */
{                                        /* Saída     : não retorna nada*/
    double media;

   media = ( v1 + v2 ) / 2.0;
   printf ("Média = %lf \n", media);     /* Exibe o resultado na tela   */
}


main ( )
{
    int  n1, n2;
    printf ("Entre 2 números inteiros: ");
    scanf  ("%d %d",&n1, &n2);
    exibe_media ( n1, n2 );              /* Chama a procedure media */
}
```

Sub-Rotinas na Linguagem "C" : Procedures e Funções

- Exemplos de sub-rotinas : *passagem de parâmetros por valor*

```
void exibe_media ( v1, v2 )                 /* Rotina:     exibe_media            */
int v1, v2;                                 /* Entrada:  v1, v2 - inteiros        */
{                                           /* Passagem de params. por valor  */
    double media;                       /* v1 e v2 recebem uma cópia de n1 e n2 */

   media = ( v1 + v2 ) / 2.0;
   printf ("Média = %lf \n", media);        /* Exibe o resultado na tela          */
   v1 = v2 = 0;                             /* Zera v1 e v2... Não afeta n1, n2 */
}


main ( )
{
    int  n1, n2;
    printf ("Entre 2 números inteiros: ");
    scanf  ("%d %d",&n1, &n2);
    exibe_media ( n1, n2 );                  /* Chama a procedure media */
}
```

Sub-Rotinas na Linguagem "C" : Procedures e Funções

- Exemplos de sub-rotinas : *passagem de parâmetros por referência (ponteiro)*

```
void calcula_media ( v1, v2, media )      /* Rotina :    calcula_media          */
int v1, v2;                               /* Passagem de params. por valor  */
double *media;                          /* Passagem de param. por referência */
{                                         /* Media é um ponteiro para result    */
   *media = ( v1 + v2 ) / 2.0;
   printf ("Média = %lf \n", *media);     /* Exibe o resultado na tela          */
   v1 = v2 = 0;                           /* Zera v1 e v2... Não afeta n1, n2 */
}

main ( )
{
    int  n1, n2;
    double result;
    printf ("Entre 2 números inteiros: ");
    scanf  ("%d %d",&n1, &n2);
    calcula_media ( n1, n2, &result);       /* Chama a procedure media */
    printf ("Valores: %d- %d \nResultado: %lf \n", n1, n2, result);
}
```

Sub-Rotinas na Linguagem "C" : Procedures e Funções

- Exemplos de sub-rotinas : *passagem de parâmetros por referência (ponteiro)*

```
void calcula_media ( v1, v2, media )      /* Rotina :    calcula_media          */
int *v1, *v2;                             /* Passagem de params. por referência */
double *media;                            /* Passagem de param.  por referência */
{
   *media = ( *v1 + *v2 ) / 2.0;
   printf ("Média = %lf \n", *media);      /* Exibe o resultado na tela          */
   *v1 = *v2 = 0;                          /* Zera v1 e v2... AFETA n1, n2 ! */
}

main ( )
{
    int  n1, n2;
    double result;
    printf ("Entre 2 números inteiros: ");
    scanf  ("%d %d",&n1, &n2);
    calcula_media ( &n1, &n2, &result);    /* Chama a procedure media */
    printf ("Valores: %d - %d \n Resultado: %lf \n", n1, n2, result);
}
```

Manipulando arquivos na Linguagem "C" :

Variável do tipo arquivo (ponteiro)...

FILE *<*nome_variável*>; /* Arquivos já existentes: stdin, stdout, stderr */

Procedimentos de manipulação de arquivos.... #include <stdio.h>

***fopen* - Abre um arquivo para leitura, escrita ou alteração**
***fclose* - Fecha o arquivo**
***fprintf* - Escreve dados em um arquivo**
***fscanf* - Lê dados de um arquivo**
***feof* - Testa para ver se encontramos o EoF (End-of-File)**

FILE *fopen (<nome_arquivo>, <modo_abertura>)
Exemplo: FILE *arquivo; arquivo = fopen ("c:\arquivo.txt","rt");
Parâmetros: modo_abertura => r (read), w (write), a (append), + (r/w)
b(binário), t (texto). Exemplos: "rt", "wb", "r+" ...
Retorno: NULL se ocorrer um erro na abertura, !(NULL) se funcionar

int fclose (FILE *<nome_variável>)
Exemplo: fclose (arquivo);

Manipulando arquivos na Linguagem “C” :

Exemplo de programa de manipulação de arquivos...

```
#include <stdio.h>

FILE *arqtxt;

main ( )
{
    char palavra[256];

    arqtxt = fopen ( “arquivo.txt”, “rt”);
    if (arqtxt == NULL)
    {
        printf (“ERRO!\n”);
        exit(1);
    }

    while ( ! feof ( arqtxt ) )
    {
       fscanf (arqtxt, “%s”, palavra);
       printf (“%s ”,palavra);
    }

   fclose (arqtxt);
}
```

```
arqtxt = fopen ( “arquivo.txt”, “wt”);

for ( ; ; )
{
    scanf  (“%s”,palavra);
    fprintf(arqtxt, “%s “, palavra);
    if (strcmp(palavra,”FIM”) == 0)
       break;
}
```

Manipulando arquivos na Linguagem “C” :

Outras rotinas de manipulação de arquivos...

fgets **- Lê uma string terminada por um ‘\n’. Acrescenta um ‘\0’ na string.**
Exemplo: fgets (<var_texto> , <tamanho_maximo>, <arquivo>);
fgets (linha, 100, arqtxt);

fputs **- Escreve uma string, substituindo o ‘\0’ por um ‘\n’.**
Exemplo: fputs (<var_texto>, <arquivo >);
fputs (linha, arqtxt);

- As rotinas fgets e fputs são muito úteis em conjunto com as rotinas sscanf e sprintf respectivamente.