

Campos e Registros: Aula Prática

6897/1 e 5187/31 – Organização e Recuperação de Dados

Profa. Valéria D. Feltrim

UEM – CTC – DIN

Slides preparados com base no Cap. 4 do livro FOLK, M.J. & ZOELLICK, B. *File Structures*. 2nd Edition, Addison-Wesley Publishing Company, 1992

Escrita de um arquivo contendo uma sequência de campos delimitados por “|”

Funções stdio.h:

- gets (str)
- printf (...)
- fopen (arq, mod)
- fputs (str, arq)
- fclose (arq)
- getchar ()

Função string.h

- strlen (str)

Função stdlib.h

- exit(1)

```
PROGRAM: writstrm
get output file name and open it with the logical name OUTPUT
get LAST name as input
while length(LAST) > 0 do
    get FIRST name, ADDRESS, CITY, STATE and ZIP as input
    write LAST to the file OUTPUT
    write "|" to the file OUTPUT
    write FIRST to the file OUTPUT
    write "|" to the file OUTPUT
    write ADDRESS to the file OUTPUT
    write "|" to the file OUTPUT
    write CITY to the file OUTPUT
    write "|" to the file OUTPUT
    write STATE to the file OUTPUT
    write "|" to the file OUTPUT
    write ZIP to the file OUTPUT
    write "|" to the file OUTPUT
    get LAST name as input
end /* while */
close OUTPUT
end PROGRAM
```

Leitura de um arquivo contendo uma sequência de campos delimitados por “|”

Funções stdio.h:

- gets (str)
- printf (...)
- fopen (arq, mod)
- fclose (arq)
- fgetc (arq)

Função stdlib.h

- exit(1)

```

PROGRAM: readstrm
get input file name and open it with the logical name INPUT
FIELD_COUNT := 0;
FIELD_LENGTH := readfield(INPUT, STR)
while (FIELD_LENGTH > 0) do
    FIELD_COUNT := FIELD_COUNT + 1
    write FIELD_COUNT and STR to the screen
    FIELD_LENGTH := readfield(INPUT, STR)
end /* while */
close INPUT
end PROGRAM

FUNCTION: readfield (INPUT, STR)
I := 0
read a character from INPUT into CH
while (not EOF(INPUT) and CH does not equal '|') do
    STR[I] := CH
    I := I + 1
    read a character from INPUT into CH
end /* while */
append '\0' to STR    /* finalização de string em C*/
return I              /* I armazena o comprimento do campo lido */
end FUNCTION
  
```

Escrita de um arquivo
contendo registros
de tamanho variável
com indicação de
tamanho no início
do registro e
campos delimitados
por “|”

Funções string.h

- strlen (str)
- strcat (str1, str2)

Funções stdio.h:

- gets (str)
- printf (...)
- fopen (arq, mod)
- fclose (arq)
- fwrite (buffer, tam_elem, qtd_elem, arq)

Pseudo writrec

```
PROGRAM: writrec

get output file name and open it with the logical name OUTPUT
get LAST name as input
while length(LAST) > 0 do
    set length of string in BUFFER to zero /* BUFFER[0] = '\0' */
    concatenate BUFFER + LAST + "|"
    for each field of record do
        get the field as input
        concatenate BUFFER + field + "|"
    end /* for */
    compute REC_LENGTH as length of string in BUFFER
    write REC_LENGTH to the file OUTPUT
    write the string in BUFFER to the file OUTPUT
    get LAST name as input
end /* while */
close OUTPUT
end PROGRAM
```

Leitura de um arquivo contendo registros de tamanho variável com indicação de tamanho no início do registro e campos delimitados por “|”

Inclui o cabeçalho **readrf.h** → Opcional

Função **string.h**:

- strtok(string_de_busca, string_delimitadora) → na 1ª chamada
- strtok(NULL, string_delimitadora) → nas chamadas subsequentes

Funções **stdio.h**:

- gets (str)
- fopen (arq, mod)
- fclose (arq)

```
PROGRAM: readrec
get input file name and open it with the logical name INPUT
set SCAN_POS to 0
REC_LENGTH := get_rec(INPUT, BUFFER)
while (REC_LENGTH > 0) do
    SCAN_POS := get_fld (FIELD, BUFFER, SCAN_POS, REC_LENGTH)
    while (SCAN_POS > 0) do
        print FIELD on the screen
        SCAN_POS := get_fld(FIELD, BUFFER, SCAN_POS, REC_LENGTH)
    end /* while */
    REC_LENGTH := get_rec(INPUT, BUFFER)
end /* while */
close INPUT
end PROGRAM
```

getrf.h:

Funções auxiliares
para a leitura
de registros e
campos de
tamanho variável

get_rec → Lê
um registro do
arquivo INPUT
para a *string*
BUFFER

get_fld → Opcional
Lê um campo
a partir da string
BUFFER

Função da stdio.h:

- fread (buffer, tam_elem, qtd_elem, arq)

```

HEADER: getrf

FUNCTION: get_rec (INPUT, BUFFER)
if EOF(INPUT) then return 0
read REC_LENGTH from INPUT
read the record contents into BUFFER
return REC_LENGTH
end FUNCTION

//A FUNÇÃO ABAIXO get_fld É OPCIONAL
FUNCTION: get_fld (FIELD, BUFFER, SCAN_POS, REC_LENGTH)
if SCAN_POS == REC_LENGTH then return 0
get a character CH at the SCAN_POS in the BUFFER
set F_POS to 0
while (SCAN_POS < REC_LENGTH and CH does not equal '|') do
    FIELD[F_POS] := CH /* place CH into FIELD*/
    F_POS := F_POS + 1;
    SCAN_POS := SCAN_POS + 1;
    get a character CH at the SCAN_POS in the BUFFER
end /* while */
append '\\0' to FIELD/* finalização de string em C*/
return SCAN_POS
end FUNCTION

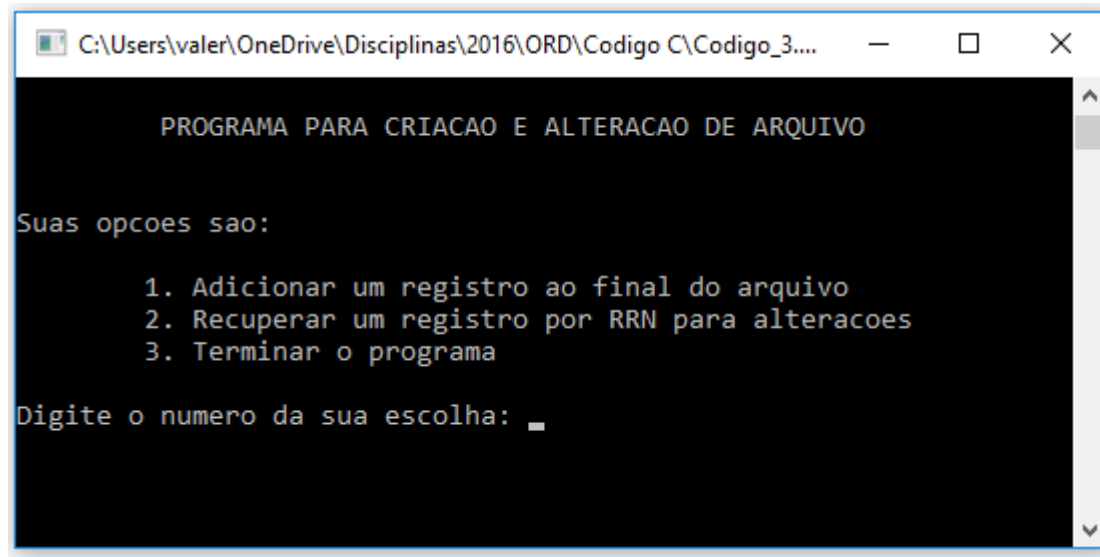
```

Busca sequencial
em um arquivo de
registros de
tamanho variável
no formato
gravado pelo
programa
writerec

Pseudo **finc.c**

```
PROGRAM: find
get input file name and open it with the logical name INPUT
get SEARCH_KEY as input /* SEARCH_KEY is a last name */
set flag MATCHED to false
while (not MATCHED and (REC_LENGTH := get_rec(INPUT, BUFFER)) > 0) do
    LAST := strtok(BUFFER, "|")
    if (LAST = SEARCH_KEY) then
        MATCHED := true
    end /* if */
end /* while */
if (MATCHED) then
    print LAST on the screen
    FIELD := strtok(NULL, "|")
    while (FIELD ≠ NULL) do
        print FIELD on the screen
        FIELD := strtok(NULL, "|")
    end /* while */
end /* if */
close INPUT
end PROGRAM
```

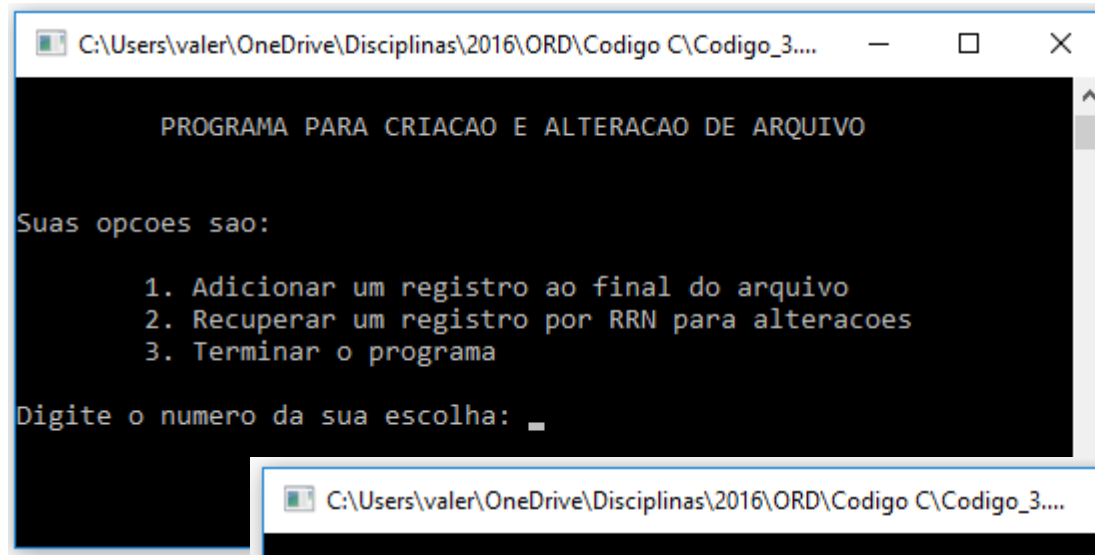
Exercício update.c



A screenshot of a Windows command prompt window. The title bar shows the file path: C:\Users\valer\OneDrive\Disciplinas\2016\ORD\Codigo C\Codigo_3.... The window has standard minimize, maximize, and close buttons. The command prompt itself has a black background with white text. It displays the title 'PROGRAMA PARA CRIACAO E ALTERACAO DE ARQUIVO', followed by 'Suas opcoes sao:', then a numbered list of three options: '1. Adicionar um registro ao final do arquivo', '2. Recuperar um registro por RRN para alteracoes', and '3. Terminar o programa'. At the bottom, it prompts 'Digite o numero da sua escolha:' followed by a cursor and a space.

```
C:\Users\valer\OneDrive\Disciplinas\2016\ORD\Codigo C\Codigo_3....  
  
PROGRAMA PARA CRIACAO E ALTERACAO DE ARQUIVO  
  
Suas opcoes sao:  
  
    1. Adicionar um registro ao final do arquivo  
    2. Recuperar um registro por RRN para alteracoes  
    3. Terminar o programa  
  
Digite o numero da sua escolha: _
```


Exercício update.c



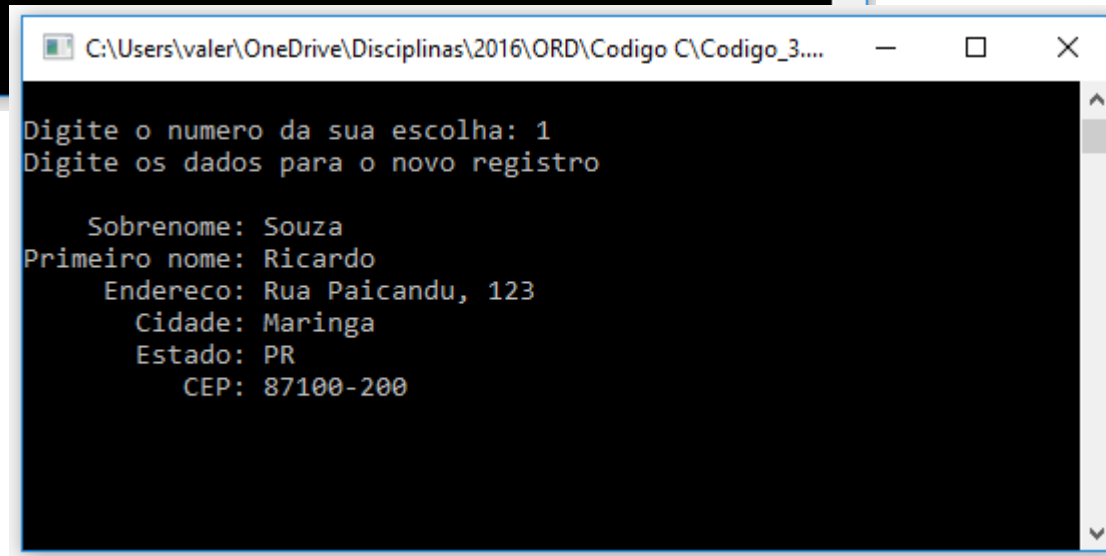
```
C:\Users\valer\OneDrive\Disciplinas\2016\ORD\Codigo C\Codigo_3....
```

```
PROGRAMA PARA CRIACAO E ALTERACAO DE ARQUIVO
```

Suas opcoes sao:

1. Adicionar um registro ao final do arquivo
2. Recuperar um registro por RRN para alteracoes
3. Terminar o programa

Digite o numero da sua escolha: _



```
C:\Users\valer\OneDrive\Disciplinas\2016\ORD\Codigo C\Codigo_3....
```

```
Digite o numero da sua escolha: 1
Digite os dados para o novo registro

Sobrenome: Souza
Primeiro nome: Ricardo
Endereco: Rua Paicandu, 123
Cidade: Maringa
Estado: PR
CEP: 87100-200
```

Exercício update.c

```
C:\Users\valer\OneDrive\Disciplinas\2016\ORD\Codigo C\Codigo_3....  -  □  X

PROGRAMA PARA CRIACAO E ALTERACAO DE ARQUIVO

Suas opcoes sao:

    1. Adicionar um registro ao final do arquivo
    2. Recuperar um registro por RRN para alteracoes
    3. Terminar o programa

Digite o numero da sua escolha: _
```

```
C:\Users\valer\OneDrive\Disciplinas\2016\ORD\Codigo C\Codigo_3.Camp...  -  □  X

Digite o RRN do registro: 0

Conteudo do registro
    feltrim
    valeria
    rua x
    mga
    pr
    87100

Voce quer modificar este registro?
    Responda S or N, seguido de <ENTER> ==>_
```

Exercício update.c

```
C:\Users\valer\OneDrive\Disciplinas\2016\ORD\Codigo C\Codigo_3.Camp...
Digite o RRN do registro: 0

Conteudo do registro
    feltrim
    valeria
    rua x
    mga
    pr
    87100

Voce quer modificar este registro?
Responda S or N, seguido de <ENTER> ==>_
```

```
C:\Users\valer\OneDrive\Disciplinas\2016\ORD\Codigo C\Codigo_3.Camp...
Voce quer modificar este registro?
Responda S or N, seguido de <ENTER> ==>s

Digite os novos dados do registro:

    Sobrenome: _
```

Estrutura e tamanho dos registros

■ Registros de tamanho fixo com campos de tamanho variável

- O registro de tamanho fixo serve como um *container* para campos de tamanho variável
- Diminuição dos problemas de espaço em comparação ao uso de campos de tamanho fixo
 - O tamanho fixo do registro pode ser uma média dos possíveis tamanhos de campo
- É preciso sinalizar de alguma forma onde os dados de cada registro terminam e onde começam as sobras de espaço
 - Qualquer técnica para delimitação de registro vista anteriormente vai funcionar
 - Alternativa → preencher o espaço vazio com algum caracter sinalizador (p.e., '\0' em C)

64 bytes

Silva Alan Rua Tiete 123 Maringa PR 87100
Flores Andre Rua Braga 34 Sarandi PR 87111

Programa que abre ou cria um arquivo de **registros de tamanho fixo**

Os registros podem ser inseridos ou lidos.

Registros a serem lidos são **buscados por RRN**

Use um buffer de tamanho **64+1** → todo reg ocupará 64 bytes, portanto, limpe o buffer antes de cada leitura

Armazene o número total de regs no cabeçalho do arquivo

O cabeçalho deve ser gravado já na criação do arquivo e deve ser lido/atualizado a cada utilização

```
PROGRAMA: update
Leia o nome do arquivo em filename
Abra o arquivo filemane para L/E:
    Se o arq não existir:
        crie e abra o arquivo para L/E
        faça header.reg_cont = 0 e grave-o no arquivo
    Senão:
        leia o cabeçalho (header) e armazene-o em header.reg_cont
Leia a opção do usuário /*(1)inserir (2)buscar/atualizar (3)sair */
Enquanto (opção < 3) faça
    Caso opção == 1:
        leia os dados do registro /* sobrenome, nome, etc */
        concatene os dados no buffer com os delimitadores
        calcule a posição de gravação //header.reg_cont*64+sizeof(header)
        faça o seek para a posição correta e grave o registro
        incremente header.reg_cont
    Caso opção == 2
        leia o RRN a ser buscado
        se RRN >= header.reg_cont imprima msg de erro e saia do Caso
        calcule a posição de leitura // RRN * 64 + sizeof(header)
        faça o seek, leia o registro para o buffer e mostre na tela
        Leia a opção do usuário
Fim enquanto
Faça seek para o início do arquivo
Grave o cabeçalho
Feche o arquivo
```