

Painel ► Cursos ► INE5412-04208A/B (20232) ►

Unidade 4 - Sistemas de Arquivos e Entrada/Saída ► Q 4.1 - Manipulação de arquivos em C

Iniciado em	Friday, 6 Oct 2023, 18:46
Estado	Finalizada
Concluída em	Sunday, 8 Oct 2023, 20:36
Tempo empregado	2 dias 1 hora
Notas	1,67/2,00
Avaliar	8,33 de um máximo de 10,00(83%)

Questão 1

Parcialmente
correto

Atingiu 0,67 de
1,00

Um arquivo **binário** contém **vários** dados do tipo **coordenada_t**, conforme definido abaixo:

```
typedef struct {
    unsigned long int id;
    int x;
    int y;
} coordenada_t;
```

Com base na estrutura definida anteriormente, o professor da disciplina de Sistemas Operacionais pediu aos alunos que implementassem uma função que busca uma **coordenada_t** no arquivo, utilizando-se para isso o seu identificador (**id**), e atualiza os seus dados. A solução proposta por um aluno é mostrada abaixo:

```
1 void atualiza_coordenada(coordenada_t antiga, coordenada_t nova,
char *nome_arquivo) {
2     FILE *f;
3     coordenada_t item;
4     f = fopen(nome_arquivo, "r+b");
5     while((fread(&item, sizeof(coordenada_t), 1, f) > 0) && (item.id
!= antiga.id));
6     if (item.id == antiga.id)
7         fwrite(&nova, sizeof(coordenada_t), 1, f);
8     fclose(f);
9 }
```

Marque as afirmativas **corretas** sobre a solução proposta pelo aluno.

ATENÇÃO: Cada resposta incorreta anula uma resposta correta.

Escolha uma ou mais:

- ☐ a. Na linha 5, a função **fread** poderia ser alterada para **fread(&item, sizeof(char), sizeof(coordenada_t), f)** sem alterar o comportamento do programa.
- ☒ b. Na linha 4, a abertura do arquivo é realizada com modo **"r+b"**, pois assim é possível realizar a atualização da **coordenada_t** antiga. ✓
- ☒ c. Seria necessário incluir a linha **fseek(f, -1*sizeof(coordenada_t), SEEK_CUR)** ; entre as linhas 6 e 7 para que a solução estivesse 100% correta. ✓
- ☐ d. A linha 6 poderia ser removida sem nenhum tipo de impacto no comportamento da solução.
- ☐ e. O arquivo poderia também ter sido aberto com modo **"wb"**, sem alterar em nada o comportamento do programa.

Sua resposta está parcialmente correta.

Você selecionou corretamente 2.

As respostas corretas são: Na linha 4, a abertura do arquivo é realizada com modo **"r+b"**, pois assim é possível realizar a atualização da **coordenada_t** antiga., Seria necessário incluir a linha **fseek(f, -1*sizeof(coordenada_t), SEEK_CUR)** ; entre as linhas 6 e 7 para que a solução estivesse 100% correta., Na linha 5, a

função `fread` poderia ser alterada para `fread(&item, sizeof(char),
sizeof(coordenada_t), f)` sem alterar o comportamento do programa.

Questão 2

Correto

Atingiu 1,00 de
1,00

Considere um arquivo texto mostrado a seguir:

```
34
0123
3456
6789
```

Os dados armazenados no arquivo se referem a uma matriz. Na primeira linha, é especificado o número de linhas (neste exemplo, 3) e de colunas (neste exemplo, 4) da matriz. Então, nas linhas seguintes são armazenados os valores da matriz (neste exemplo, uma matriz de 3 linhas e 4 colunas). Considere que os números armazenados no arquivo contêm somente um dígito (isto é, números de 0 até 9).

O programa inicializa uma matriz `m` com os valores lidos do arquivo de entrada `matriz.txt`:

```
FILE *f;
f = fopen("matriz.txt", "r");
int linhas = lenumero(f);
int colunas = lenumero(f);
int **m = (int **) malloc(sizeof(int *) * linhas);
for(int i = 0; i < linhas; i++) {
    m[i] = (int *) malloc(sizeof(int) * colunas);
    for(int j = 0; j < colunas; j++)
        m[i][j] = lenumero(f);
}
fclose(f);
```

Selecione a alternativa que implementa **corretamente** a função `lenumero()` utilizada no código anterior.

Escolha uma opção:

☐ a.

```
int lenumero(FILE *f) {
    int numero = fgetc(f);
    if(numero != '\n')
        numero = numero - '0';
    else
        fgetc(f);
    return numero;
}
```

☐ b.

```
int lenumero(FILE *f) {
    int numero = fgetc(f);
    if(numero != '\n')
        numero = numero;
    else
        fgetc(f);
    return numero;
}
```



c.

```
int lenumero(FILE *f) {
    int numero = fgetc(f);
    return numero - '0';
}
```



d.

```
int lenumero(FILE *f) {
    int numero = fgetc(f);
    if(numero != '\n')
        return numero - '0';
    return fgetc(f) - '0';
}
```



e.

```
int lenumero(FILE *f) {
    int numero = fgetc(f);
    if(numero != '\n')
        return numero;
    return fgetc(f);
}
```



f.

```
int lenumero(FILE *f) {
    return fgetc(f);
}
```

Sua resposta está correta.

A resposta correta é:

```
int lenumero(FILE *f) {
    int numero = fgetc(f);
    if(numero != '\n')
        return numero - '0';
    return fgetc(f) - '0';
}
```

◀ Entrada/Saída: RAID

Seguir para...



Q 4.2 - Sistemas de arquivos e otimização ▶