



UFMS Universidade Federal de Mato Grosso do Sul



Algoritmos e Programação II
Prof. Dr. Edson Takashi Matsubara
Monitor Anderson Bessa
17 de Janeiro de 2013

Trabalho Prático **Hotel**

1. Introdução

Neste trabalho você implementará um sistema para gerenciar um hotel. O objetivo consiste em simular antigos sistemas para reservas de hotel, utilizando dos conceitos vistos na disciplina de Algoritmos e Programação II.

2. Descrição do Trabalho

O sistema proposto deverá prover as seguintes funcionalidades básicas:

- *Listar quartos disponíveis e ocupados:* Serão exibidos todos os quartos juntamente com suas informações;
- *Check-in:* Serão fornecidos como parâmetro o nome da pessoa (adote apenas o primeiro nome), a data atual (adote o formato `yyyymmdd`) e o tipo de quarto. Deverá ser realizado uma busca para verificar se existe o tipo de quarto de hotel disponível. Caso verdadeiro, imprima uma mensagem de sucesso e retorne 1. Caso contrário retorne uma mensagem informando que não existe o tipo de quarto disponível e retorne 0;
- *Check-out:* Será fornecido como parâmetro apenas o nome da pessoa. Deverá ser realizado uma busca para verificar se esta pessoa realmente encontra-se no hotel. Caso verdadeiro, imprima uma mensagem indicando que o quarto foi liberado e retorne 1. Caso contrário, imprima uma mensagem indicando que o hóspede não pode ser encontrado e retorne 0;
- Pesquisar `<OP> <ID>`: Onde OP pode ser “pessoa” ou “quarto”. Deverá ser realizada uma **BUSCA BINÁRIA** no **ARQUIVO DE ÍNDICES** (explicado posteriormente);

A informação sobre um quarto de hotel deverá ser armazenada em um registro (**struct**), conforme apresentado abaixo;

```
typedef struct _quarto{
    char nome[30];           // nome do hospede
    int numero_quarto;
    int data_check_in;
    int data_check_out;
    int tipo_quarto;
    int ocupado;
}quarto;
```

Como restrição, o hotel possuirá 10 quartos duplos (tipo 2), numerados de 1 a 10 e 5 quartos triplos (tipo 3), numerados de 11 a 15.

O registro (um para cada quarto) deverá ser armazenado em um arquivo binário chamado *hotel.bin* (referência em <http://www.cplusplus.com/reference/cstdio/>). Para cada quarto de hotel, deverá haver um registro. O arquivo deverá ser em modo binário, pois solicitado que se faça um acesso direto a um registro em específico.

Deverá ser criado um arquivo de índices com o nome dos hóspedes. Um arquivo de índice será um arquivo de texto (.txt), onde cada linha irá conter o nome de um hóspede (**ORDENADO EM ORDEM CRESCENTE**) e o índice do registro, separado por um espaço. Assim, quando for realizado a pesquisa por um hóspede, deverá ser realizado uma **BUSCA BINÁRIA** no arquivo de índices e obter o índice do registro que contém a informação do hóspede. Deve-se então realizar um acesso direto no Arquivo de Registros (dica: utilize a função *fseek*, com o tamanho de *sizeof (quarto) * índice*).

Deverão ser implementados dois arquivos de índices: *nomes.txt* e *quartos.txt* respectivamente representando os campos NOME e QUARTO.

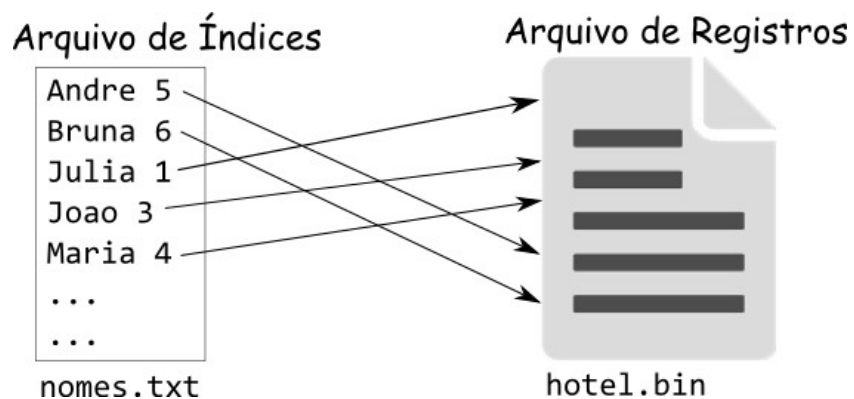


Figure Relação entre o arquivo de índices e o arquivo de registros

3. Funcionamento

Seu sistema deverá conter pelo menos quatro executáveis que utilizam “argc” e “argv” para obter os parâmetros da execução.

- Listar informação de todos os quartos

```
$ ./listar_quartos
```

- Pesquisar por hóspede/quarto

```
$ ./pesquisar pessoa Joao  
$ ./pesquisar quarto 10
```

- Realizar check-in

```
$ ./check_in Joao 20120117 2
```

- Realizar check-out

```
$ ./check_out Joao
```

4. **Entrega do Trabalho**

O trabalho deverá ser enviado pelo moodle até às 23h30min **do dia 10 de Fevereiro**. O código fonte deverá estar documentado. O trabalho deverá ser feito individualmente. Casos de plágio serão tratados com rigor. Identifique-se com seu nome nos arquivos fontes do trabalho. Será descontado 1 ponto a cada dia de atraso na entrega do trabalho.

5. **Organização do Código**

Seu código fonte deverá utilizar “header files” (.h) e todas as funções pertinentes as funcionalidades do hotel deverão estar nestes dois arquivos:

hotel.c (código fonte das funções necessárias para o trabalho)
hotel.h (arquivo de cabeçalho (header files))

que serão chamados pelos arquivos:

check_in.c
check_out.c
pesquisa.c
listar_quartos.c

Que deverão conter #include “hotel.h” para realizar as chamadas necessárias para a execução do sistema do hotel. Seu código deverá ser compilado utilizando makefile.

6. **Avaliação**

Será levando em conta a qualidade do código em termos de legibilidade (código fácil de entender - evite coisas muito diferentes do usual), reuso (seu código pode facilmente ser utilizado em outro programa? Lembre-se do que foi comentado em sala sobre a interface Programa <-> Dados), organização (uso correto do header file), apresentação (saídas bem formatadas e de fácil compreensão), comentário e cabeçalhos de cada função, entre outros.

Além da correção do programa, o professor poderá questionar o autor do trabalho sobre assuntos pertinentes.