Atividade I

Linguagem de Programação I Instituto Metrópole Digital

2020.1

Quando eu escrevi isto?

Esta semana vimos como funcionam os *streams* de entrada e saída no C++. Utilizamos eles para escrevermos e lermos dados do terminal e em arquivos. Começamos também a criar um programa, que chamaremos de `diary` neste documento, que armazena e lista mensagens. Atualmente, ele possui a seguinte interface de utilização:

- `./diary`
  - Exibe uma mensagem informando como o programa deve ser utilizado.
- `./diary add`
  - Solicita através do `std::cin` uma mensagem ao usuário e adiciona-a em uma linha do arquivo.
- `./diary add <mensagem>`
  - Adiciona `<mensagem>` em uma linha do arquivo.
- `./diary list`
  - Lista todas as mensagens salvas no arquivo.

Uma característica importante em programas similares é o armazenamento da data e hora em que uma mensagem foi adicionada. O objetivo desta atividade é adicionar esta funcionalidade ao nosso programa.

Descrição da Atividade

O objetivo desta atividade é vocês agruparem mensagens pelo dia que ela foi cadastrada e adicionarem a hora que a mensagem foi cadastrada como prefixo em cada linha.

O formato que utilizaremos no texto será o **markdown** pela sua simplicidade e facilidade de exportação para outros formatos. O arquivo que armazena as mensagens deve possuir o seguinte formato:

- Para cada dia, deverá ser criado um **título**. Para criar um título em *markdown*, utilizamos uma cerquilha no início da linha (#). Exemplo:

```
# 17/06/2020
<mensagens do dia 17/06/2020>

# 18/06/2020
<mensagens do dia 18/06/2020>

# 19/06/2020
<mensagens do dia 19/06/2020>
...
```

- Cada mensagem deve ser precedida do horário em que foi cadastrada no formato **<horas>:<minutos>:<segundos>**, e devem ser colocadas em uma lista não ordenada. Para fazer isto com *markdown*, é necessário colocar um traço no início da linha (-). Exemplo:

```
# 17/06/2020

- 08:00:00 <mensagem 1>
- 15:33:25 <mensagem 2>
- 19:22:03 <mensagem 3>

# 18/06/2020

- 08:00:00 <mensagem 4>
- 15:33:25 <mensagem 5>
- 19:22:03 <mensagem 6>

# 19/06/2020

- 08:00:00 <mensagem 7>
- 15:33:25 <mensagem 8>
- 19:22:03 <mensagem 9>
```

Linhas em branco podem ser utilizadas para organizar o arquivo como desejarem. Só não são permitidas entre as mensagens de um dia, pois isto criaria várias listas ao invés de uma só.

Data e hora no *C++*

É necessário utilizar o pacote a função `std::strftime` do pacote `ctime`:

```cpp
#include <ctime>
#include <string>

std::string format_current_date(const std::string& format)
{
    std::time_t time = std::time(nullptr);
    char result[1024];

    std::strftime(result, sizeof(result), format.c_str(), std::localtime(&time));

    return std::string(result);
}

std::string get_current_date()
{
    return format_current_date("%d/%m/%Y");
}

std::string get_current_time()
{
    return format_current_date("%H:%M:%S");
}
```

Detalhes de Implementação

- Você deve verificar se a data de hoje já existe no arquivo e só adicionar um novo título se ela não existir.
- O arquivo está ordenado em ordem crescente. Então, você sempre irá adicionar suas mensagens no final do arquivo.
- Você pode acessar todos os trechos de código produzido durantes as aulas no seguinte link: **http://waldson.com.br:3000/waldson/lp1-2020.5**
- Lembrem-se que o `list` deve retornar apenas as mensagens e não o conteúdo do arquivo completo

Entrega

- O exercício deve ser entregue em um arquivo zip.
- Não será tolerado qualquer tipo de compartilhamento de trabalho entre os alunos. Tal ação resulta em anulação da pontuação para **TODOS** os envolvidos. Ou seja, colou ou deu cola? Zero na nota.
- **NÃO** serão aceitos envios por e-mail. O arquivo deve ser enviado **apenas** pelo *SIGAA* através da opção *Tarefas* até a data divulgada no sistema.