

# Atividade III

Linguagem de Programação I      Instituto Metrópole Digital  
2020.5

## Introdução

Nesta semana aprendemos sobre ponteiros, contêineres STL, e também sobre como organizar ainda mais nossos programas para perdemos menos tempo na compilação e manutenção dele, além de deixá-los com uma estrutura mais profissional.

Nesta atividade utilizaremos todos estes conhecimentos para consolidarmos alguns desses tópicos.

## Descrição da Atividade

A atividade consiste na implementação de 2 funcionalidades totalizando 5 pontos:

- **Carregar configurações de execução a partir de um arquivo (2,0 pontos)**  
O caminho do diário deve vir de um arquivo de configuração.
- **Deixar o usuário decidir como a mensagem deve ser exibida (1,0 ponto)**  
O usuário poderá escolher como a mensagem deve ser exibida.
- **Uso interativo (2,0 pontos)**  
O programa deve possuir o novo modo de utilização através de menus interativos

## Carregar Configurações a Partir de um Arquivo

Até o momento, nosso programa de diário está compilado com alguns parâmetros que deveriam ser configurações:

1. **O caminho do arquivo do diário:** se o usuário quiser mudar a localização ou o nome do arquivo, é necessário possuir o código fonte do programa, entender onde deve alterar isso e recompilá-lo. Isto é algo impraticável do ponto de vista de desenvolvimento de software.
2. **A forma que as mensagens são impressas por padrão** `./diary list:` com esta funcionalidade, deixamos que o usuário escolha o formato que ele quer exibir a mensagem, seja só a mensagem em si, ou ela com a data e a hora.

## Arquivo de configuração

Para esta atividade, o arquivo de configuração deverá estar localizado na mesma pasta do executável e se chamar `diary.config`. Ele deve conter duas configurações, como no exemplo abaixo:

```
path=<caminho para o arquivo de diário>
default_format=<formato que a mensagem deve ser exibida>
```

Por exemplo:

```
path=/home/aluno/.config/diary.md
default_format=[%d %t] %m
```

onde:

- O programa lerá e escreverá as mensagens em `/home/aluno/.config/diary.md`
  - As mensagens listadas devem utilizar o formato e fazer substituições:
    - `%d` deve ser substituído pela data da mensagem
    - `%t` deve ser substituído pela hora da mensagem
    - `%m` deve ser substituído pelo texto da mensagem
    - Todos os outros caracteres devem ser impressos do jeito que estão no formato
- \* Exemplo de saída do formato anterior: `[01/07/2020 09:35:16] Mensagem`

**Importante:** se o arquivo de configuração não existir, um deve ser criado com o seguinte conteúdo:

```
path=diary.md
default_format=%d %t: %m
```

**Atenção:** A ordem dos parâmetros de configuração não importa. Seu programa deverá ser capaz de aceitar um arquivo de configuração com os parâmetros em qualquer ordem.

## Deixar o usuário decidir como a mensagem deve ser exibida

A chamada à `./diary list` deverá aceitar um parâmetro opcional com o formato que o usuário deseja imprimir a mensagem: `./diary list <format>`. Se o parâmetro `format` não for informado, o programa deve utilizar o formato padrão do arquivo de configuração.

## Uso Interativo

O usuário deverá ser capaz de utilizar o programa através de um menu interativo:

Diário 1.0

Selecione uma ação:

- 1) Listar mensagens
- 2) Adicionar nova mensagem
  
- 0) Finalizar

Este modo de uso deverá ser inicializado ao chamar `./diary interactive` e deve continuar rodando enquanto o usuário não finalizar (escolher a opção zero ou matar o programa). As funcionalidades para as ações devem ser reaproveitadas do código que vocês já possuem.

## Organização do Projeto

Vimos durante a semana, formas de organizar melhor nosso projeto e isto também contará na nota final da avaliação. Projetos desorganizados serão penalizados. Alguns itens que custaram pontos na avaliação do projeto:

- Projeto sem Makefile
- Makefile sem a regra `clean`
- Cabeçalhos fora da pasta `include`
- Arquivos de implementação fora da pasta `src`
- Projeto sem README.md
- Especificações do programa diferente das listadas na atividade (Exemplo: `./prog interativo` ao invés de `./prog interactive`)

## Entrega

- Trabalhos atrasados poderão serem enviados, via Discord, para mim até 06/07/2020 às 23:59:59, custando 0,5 ponto por dia de atraso.
- 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.