

Aula Prática 1 - Roteiro

27/07/2021 - Roteiro referente à aula prática 1 - Uso das funções `printf` e `strlen`, da palavra-chave `sizeof` e do operador ternário. Exibir um nome recebido através da interface de linha de comando e os tamanhos dos tipos básicos, dos modificadores de sinal, dos modificadores de largura e das possíveis combinações. Uso de arquivos de dependências e de um sistema de controle de versão (RCS - Revision Control System).

Versões:

- **27/07/2021 - Versão inicial**
- **29/07/2021 - Correção do roteiro com a inclusão dos rótulos `clean-objs`, `clean-gcc` e `clean-clang`.**

Prazo: 29/07/2021 - 18:00

Valor: 10,0 - Peso: 1

Observações:

- Leia este enunciado com **MUITA** atenção até o final antes de iniciar o trabalho.
- Os arquivos solicitados deverão estar disponíveis nos diretórios correspondentes (**Aulas-Práticas** e **RCS**) até o prazo estipulado acima. Cuidado com os nomes dos diretórios e dos arquivos. Deverão ser exatamente os definidos neste roteiro (maiúsculas, minúsculas, caracteres especiais e extensões, se existentes).
- **As tarefas deverão ser executadas na ordem solicitada neste roteiro.**
- Os arquivos de dependências deverão possibilitar que a compilação e a *linkedição* sejam executadas utilizando-se tanto o *gcc*, quanto o *clang*. A seleção da ferramenta utilizada deverá ser realizada no momento da execução do comando *make*. O *gcc* deverá ser considerado o valor padrão para a compilação e para a *linkedição*.

Para a definição da ferramenta desejada deverá ser utilizada uma macro (no *FreeBSD*) ou um argumento com o valor desejado (no *CentOS*). As duas macros utilizadas deverão ser *GCC* e *CLANG* (definidas usando a opção *-D*). O argumento, identificado por *cc*, deverá ser igual a *GCC* ou *CLANG*.

- Independente da ferramenta utilizada para a compilação, o *flag* de compilação deverá ser definido no instante da execução do comando *make*. O valor padrão para este *flag* deverá ser *"-Wall -ansi"* (sem as aspas).

Durante a execução do comando *make* poderão ser definidos outros valores para este *flag* (mantendo a opção de exibir todas as mensagens de advertência) através de macros ou através de argumentos (de forma semelhante àquela utilizada para definir o compilador/*linkeditor*). No *FreeBSD* deverão ser definidas as macros *ANSI*, *C89*, *C99* e *C11*, enquanto que no *CentOS* deverá ser definido o argumento *dialecto* com os valores *ANSI*, *C89*, *C99* ou *C11*.

- Crie uma macro, *DIALETO*, contendo o *dialecto* a ser utilizado na compilação do código. Esta macro será inicialmente igual a *"ansi"* e poderá ser alterada para *"c89"*, *"c99"* ou *"c11"* de acordo com o esquema definido acima.
- O *flag* de *linkedição* deverá ser igual a *"-Wall"* (sem as aspas).
- Cuidado com os nomes das macros e dos rótulos. Deverão ser exatamente os definidos neste roteiro (maiúsculas, minúsculas, caracteres especiais e extensões, se existentes).
- Todos os rótulos solicitados no roteiro são obrigatórios. Durante a correção, caso não seja possível alcançar os objetivos (binários e/ou bibliotecas e limpezas de código) solicitados, a nota correspondente ao item/aula questão será zero.
- Seguem alguns exemplos:

make - compila/linkedita (tanto no FreeBSD, quanto no CentOS) com a ferramenta e dialeto padrões, ou seja, gcc e ANSI respectivamente.
 make -DGCC - compila/linkedita usando o *gcc* e o dialeto ANSI (somente FreeBSD).
 make -DCLANG - compila/linkedita usando o *clang* e o dialeto ANSI (somente FreeBSD).
 make cc=GCC - compila/linkedita usando o *gcc* e o dialeto ANSI (somente CentOS).
 make cc=CLANG - compila/linkedita usando o *clang* e o dialeto ANSI (somente CentOS).
 make -DCLANG -DC89 - compila/linkedita usando o *clang* e o dialeto C89 (somente FreeBSD).
 make -DCLANG -DC11 - compila/linkedita usando o *clang* e o dialeto C11 (somente FreeBSD).
 make cc=CLANG dialeto=C99 - compila/linkedita usando o *clang* e o dialeto C99 (somente CentOS).

- Inclua, no início de todos os arquivos solicitados, os seguintes comentários:

```
Universidade Federal do Rio de Janeiro
Escola Politecnica
Departamento de Eletronica e de Computacao
EEL270 - Computacao II - Turma 2021/1
Prof. Marcelo Luiz Drumond Lanza
Autor: <nome completo>
Descricao: <descrição sucinta dos objetivos do programa>
```

```
$Author$
$Date$
$Log$
```

- Inclua, no final de todos os arquivos solicitados, os seguintes comentários:

```
$RCSfile$
```

Antes de começar as tarefas desta aula, leia com atenção os itens 1, 2, e 3.

1. Após se conectar remotamente à máquina loghost02, usando o *MobaXterm* ou o cliente SSH desejado, execute o comando *screen* e crie 3 terminais virtuais:
 - 1) Edição
 - 2) Compilação, linkedição e testes na máquina loghost02 (FreeBSD).
 - 3) Compilação, linkedição e testes na máquina loghost03 (CentOS).

A partir da máquina loghost02 é possível se conectar à máquina loghost03 executando o comando: "ssh loghost03" (sem as aspas).

2. Crie o diretório "*~/private/EEL270/2021-1/Aulas-Praticas/RCS*" utilizando o comando da CLI - *Command Line Interface* - apropriado (*mkdir*). Veja na página de manual do comando como criar todos os diretórios com uma única execução do comando. Os nomes dos diretórios deverão ser os indicados neste roteiro, respeitando-se os caracteres maiúsculos e/ou minúsculos.
3. Verifique qual o espaço em disco ocupado pela sua conta no momento (comando *du - sh*). Atualmente, a conta de cada usuário tem um limite máximo de 500 MBytes. O comando *du* deverá ser executado a partir do diretório raiz da sua conta.
4. Crie o arquivo "*aula0101.c*" (no diretório "Aulas-Praticas") contendo o código-fonte de um programa que receba um nome que possa conter caracteres de espaço e que tenha no máximo 50 caracteres. Este nome deverá ser recebido através de um único argumento de linha de comando. O programa deverá exibir o nome recebido delimitado por aspas e centralizado na tela do terminal (assuma que a tela do terminal tem 80 colunas). A linha contendo o nome deverá ser precedida por uma linha em branco, por uma linha com 80 caracteres '-' (hífens) e por uma linha em branco. O mesmo padrão (linha em branco, linha com 80 hífens e linha em branco) deverá ser utilizado após a linha contendo o nome.

Esta sequência (linha em branco, linha contendo 80 hífen e linha em branco) deverá ser gerada (exibida) pela função *ExibirSeparador*, que deverá ser implementada no arquivo "*aula0101.c*" (usando o laço de repetição desejado).

```
void
ExibirSeparador (void);
```

Observação:

A função `strlen` retorna o comprimento útil de uma string.

5. Crie os arquivos de dependências ("*BSDmakefile*" e "*GNUmakefile*") contendo as macros **CC**, **LD**, **CFLAGS**, **LFLAGS**, **DIALETO**, **AULA01** (correspondendo a "*aula0101*"), **AULA0101OBS** (correspondendo a "*aula0101.o*"), **EXECS** (correspondendo ao valor da macro **AULA01**), **LIBS** (vazio) e **ALL** (correspondendo aos valores das macros **EXECS** e **LIBS**). Inclua, nos arquivos de dependências, os rótulos **all**, **aula01**, **aula0101**, **clean**, **clean-all**, **clean-objs**, **clean-freebsd**, **clean-linux**, **clean-gcc**, **clean-clang**, **clean-ansi**, **clean-c89**, **clean-c99** e **clean-c11** com as declarações necessárias.

Inclua, nos comandos relativos ao objetivo *aula0101*, o comando necessário para criar uma **cópia** do executável com o nome que inclua o sistema operacional, a ferramenta de compilação/linkedição e o dialeto da linguagem C utilizado, de acordo com o exemplo abaixo:

"aula0101-FreeBSD-gcc-ansi" - versão FreeBSD, gerada usando o gcc e o padrão ANSI.

6. Gere e teste as 16 possíveis versões do executável "*aula0101*" (CentOS x FreeBSD, clang x gcc, ansi x C99 x C11).
7. Submeta os arquivos "*aula0101.c*" e "**makefile*" ao sistema de controle de versão.
8. Recupere uma cópia de leitura do arquivo "*aula0101.c*" e uma cópia de escrita dos arquivos "**makefile*".
9. Crie o arquivo "*aula0102.c*" (no diretório "Aulas-Praticas") contendo um programa que exiba o título "Tamanhos dos Tipos Basicos:". Este título deverá ser exibido após uma linha em branco, uma linha contendo 80 hífens e uma linha em branco, iniciando na primeira coluna e incluindo as aspas e o caractere dois pontos. Não poderá incluir nenhum caractere especial, como por exemplo, os acentos.

Após uma linha em branco, o programa deverá exibir os tamanhos em bytes dos cinco tipos básicos da linguagem C. O nome do tipo (exatamente igual à palavra-chave correspondente) deverá ser exibido a partir da coluna 1, enquanto que o tamanho em bytes deste tipo (utilizando-se a palavra chave *sizeof*) deverá ser exibido a partir da coluna 61. A palavra *byte* ou a palavra *bytes* (mantendo a concordância) deverá ser exibida após um caracteres de espaço colocado após o tamanho do tipo em questão.

Após o último tipo, deverá ser incluída uma linha em branco, uma linha com 80 hífens e uma linha em branco.

O argumento da função *printf* deverá usar o operador ternário para selecionar entre a palavra *byte* e a palavra *bytes*. A implementação só poderá utilizar variáveis para o controle dos laços de repetição necessários.

Operador Ternário:

Condição ? verdadeiro : falso

Onde

Condição é a condição que será testada.

Verdadeiro é o que fazer quando a condição for verdadeira.

Falso é o que fazer quando a condição for falsa.

A implementação deste programa só poderá utilizar uma variável, a necessária para o controle dos laços de repetição.

A sequência (linha em branco, linha contendo 80 hífens e linha em branco) deverá ser gerada (exibida) pela função *ExibirSeparador*, que deverá ser implementada no arquivo "*aula0102.c*" (usando o laço de repetição desejado).

void

ExibirSeparador (void);

10. Inclua, no arquivo de dependências, a macro **AULA0102OBJS** e o rótulo **aula0102** com as declarações necessárias. Atualize os valores das macros, onde for necessário.
11. Gere e teste as 16 possíveis versões do executável "aula0102".
12. Crie uma cópia do arquivo "aula0102.c" com o nome "aula0103.c".
13. Submeta os arquivos "aula0102.c" e *"*makefile"* ao sistema de controle de versão.
14. Recupere uma cópia de leitura do arquivo "aula0102.c" e uma cópia de escrita dos arquivos *"*makefile"*.
15. **Inclua**, no arquivo "aula0103.c", as linhas necessárias para exibir os tamanhos em bytes dos modificadores de sinal e dos modificadores de largura.

Antes de exibir os tamanhos dos modificadores de sinal, exiba a mensagem "Tamanhos dos Modificadores de Sinal:". Este título deverá ser exibido após uma linha em branco, uma linha contendo 80 hífen e uma linha em branco, iniciando na primeira coluna e incluindo as aspas e o caractere dois pontos. Não poderá incluir nenhum caractere especial, como por exemplo, os acentos.

Antes de exibir os tamanhos dos modificadores de largura, exiba a mensagem "Tamanhos dos Modificadores de Largura:". Este título deverá ser exibido após uma linha em branco, uma linha contendo 80 hífen e uma linha em branco, iniciando na primeira coluna e incluindo as aspas e o caractere dois pontos. Não poderá incluir nenhum caractere especial, como por exemplo, os acentos.

A implementação deste programa só poderá utilizar uma variável, a necessária para o controle dos laços de repetição.

A sequência (linha em branco, linha contendo 80 hífen e linha em branco) deverá ser gerada (exibida) pela função ExibirSeparador, que deverá ser implementada no arquivo "aula0103.c" (usando o laço de repetição desejado).

```
void  
ExibirSeparador (void);
```

16. **Inclua**, no arquivo de dependências, a macro **AULA0103OBJS** e o rótulo **aula0103** com as declarações necessárias. Atualize os valores das macros, onde for necessário.
17. Gere e teste as 16 possíveis versões do executável "aula0103".
18. Crie uma cópia do arquivo "aula0103.c" com o nome "aula0104.c".
19. Submeta os arquivos "aula0103.c" e *"*makefile"* ao sistema de controle de versão.
20. Recupere uma cópia de leitura do arquivo "aula0103.c" e uma cópia de escrita do arquivo *"*makefile"*.
21. Inclua, no arquivo "aula0104.c", as linhas necessárias para exibir os tamanhos em bytes dos modificadores de sinal combinados com os modificadores de largura.

Antes de exibir os tamanhos das combinações, exiba a mensagem "Tamanhos dos Modificadores de Sinal combinados com os Modificadores de Largura:". Este título deverá ser exibido após uma linha em branco, uma linha contendo 80 hífen e uma linha em branco, iniciando na primeira coluna e incluindo as aspas e o caractere dois pontos. Não poderá incluir nenhum caractere especial, como por exemplo, os acentos.

A implementação deste programa só poderá utilizar uma variável, a necessária para o controle dos laços de repetição.

A sequência (linha em branco, linha contendo 80 hífen e linha em branco) deverá ser gerada (exibida) pela função ExibirSeparador, que deverá ser implementada no arquivo "aula0104.c" (usando o laço de repetição desejado).

```
void  
ExibirSeparador (void);
```

22. **Inclua**, no arquivo de dependências, a macro **AULA0104OBJS** e o rótulo **aula0104** com as declarações necessárias. Atualize os valores das macros, onde for necessário.

23. Gere e teste as 16 possíveis versões do executável "aula0104".
24. Crie uma cópia do arquivo "aula0104.c" com o nome "aula0105.c".
25. Submeta os arquivos "aula0104.c" e "*makefile" ao sistema de controle de versão.
26. Recupere uma cópia de leitura do arquivo "aula0104.c" e uma cópia de escrita dos arquivos "*makefile".
27. Inclua, no arquivo "aula0105.c", as linhas necessárias para exibir os tamanhos em bytes das combinações válidas entre tipos básicos e modificadores de sinal, das combinações válidas entre tipos básicos e modificadores de largura e das combinações válidas entre tipos básicos, modificadores de sinal e modificadores de largura.

Antes de exibir os tamanhos das combinações, exiba a mensagem correspondente, ou seja, "Tamanhos dos Tipos Básicos combinados com Modificadores de Sinal :", "Tamanhos dos Tipos Básicos combinados com Modificadores de Largura :" ou "Tamanhos dos Tipos Básicos combinados com Modificadores de Sinal e com Modificadores de Largura:". Estes títulos deverão ser exibidos após uma linha em branco, uma linha contendo 80 hífens e uma linha em branco, iniciando na primeira coluna e incluindo as aspas e o caractere dois pontos. Não poderá incluir nenhum caractere especial, como por exemplo, os acentos.

A implementação deste programa só poderá utilizar uma variável, a necessária para o controle dos laços de repetição.

A sequência (linha em branco, linha contendo 80 hífens e linha em branco) deverá ser gerada (exibida) pela função ExibirSeparador, que deverá ser implementada no arquivo "aula0103.c" (usando o laço de repetição desejado).

```
void  
ExibirSeparador (void);
```

28. Inclua, no arquivo de dependências, a macro **AULA0105OBJS** e o rótulo **aula0105** com as declarações necessárias. Atualize os valores das macros, onde for necessário.
29. Gere e teste as 16 possíveis versões do executável "aula0105".
30. Submeta os arquivos "aula0105.c" e "*makefile" ao sistema de controle de versão.
31. Recupere uma cópia de leitura do arquivo "aula0105.c" e uma cópia de escrita dos arquivos "*makefile".
32. Limpe o diretório (*make clean-all*).

Sugestões de Leitura:

1. Página de manual dos comandos *mkdir* e *du*.
2. Página de manual das funções *printf* e *strlen*.
3. Capítulo sobre RCS - livro "Programação para Linux - Aprenda em 24 horas (ver Bibliografia).
4. Capítulo sobre make - livro "Programação para Linux - Aprenda em 24 horas (ver Bibliografia).