

Aula Prática 1

Objetivos:

- Uso das funções *printf* (incluindo as cores da tabela ANSI) e *strlen*, da palavra-chave *sizeof* e do operador ternário.
- Exibir um nome recebido através da interface de linha de comando.
- Exibir os tamanhos dos tipos básicos, dos modificadores de sinal, dos modificadores de largura e das possíveis combinações destes 3 grupos.
- Uso de arquivos de dependências específicos para os sistemas operacionais das famílias BSD e Linux.
- Uso de um sistema de controle de versão (RCS - *Revision Control System*).

Versão Inicial: 25/04/2023

Prazo: 27/04/2023 – 8:00

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-Praticas* e *RCS*) até o prazo estipulado acima. Cuidado com os nomes dos diretórios e dos arquivos. Deverão ser exatamente os definidos neste roteiro (incluindo 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 que 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 como o valor padrão para a ferramenta de compilação e de *linkedição*.

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

- Independente da ferramenta utilizada para a compilação, as opções de compilação poderão ser redefinidas no instante da execução do comando *make* (mantendo-se a exibição de todas as mensagens de advertência, definida pelo valor **-Wall**). O valor padrão para estas opções deverá ser **-Wall -ansi**.

Estas opções poderão ser redefinidas 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**, **C90**, **C99** e **C11**, enquanto que no *Linux* deverá ser definido o argumento **dialeto** com um dos seguintes valores **ANSI**, **C89**, **C90**, **C99** ou **C11**.

- Os arquivos de dependências deverão incluir a macro **DIALECT** contendo o dialeto a ser utilizado na compilação do código. Esta macro será inicialmente igual a **ansi** e poderá ser alterada para **c89**, **c90**, **c99** ou **c11** de acordo com o esquema definido acima.
- Os arquivos de dependências deverão incluir também a macro **STANDARD** contendo a opção de linha de comando correspondente ao dialeto selecionado. Se, por exemplo, o dialeto selecionado for o **ANSI**, esta macro deverá ser igual a **-ansi**. Por outro lado, se o dialeto for uma das outras quatro opções, esta macro deverá ser igual a **-std=CXX**, onde **XX** deverá ser substituído pelo número correspondente (se o dialeto for igual a **C89**, **XX** deverá ser igual a **89**, se o dialeto for igual a **C90**, **XX** deverá ser igual a **90** e assim por diante).
- A *linkedição* deverá utilizar a opção **-Wall**.

- 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 em questão será igual a zero.
- Seguem alguns exemplos (todos devem funcionar):
 - ***make*** - compila/*linkedita* (tanto no *FreeBSD*, quanto no *Linux*) com a ferramenta e dialeto padrões, ou seja, ***gcc*** e **ANSI** respectivamente.
 - ***make clean-all all***
 - ***make clean-all aula01***
 - ***make clean aula0101***
 - ***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 *Linux*).
 - ***make cc=CLANG*** - compila/*linkedita* usando o ***clang*** e o dialeto **ANSI** (somente *Linux*).
 - ***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 dialero=C99*** - compila/*linkedita* usando o ***clang*** e o dialeto **C99** (somente *Linux*).
 - ***make cc=GCC dialeto=C90*** - compila/*linkedita* usando o ***gcc*** e o dialeto **ANSI** (somente *Linux*).

- Inclua, no início de todos os arquivos solicitados (código-fonte e arquivos de dependências), os seguintes comentários (**sem caracteres especiais**):

```
Universidade Federal do Rio de Janeiro
Escola Politecnica
Departamento de Eletronica e de Computacao
EEL270 - Computacao II - Turma 2023/1
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, o seguinte comentário:

```
$RCSfile$
```

- Antes de começar as tarefas desta aula, leia com atenção os itens 1, 2, e 3.
1. Abra um terminal gráfico e execute o comando **screen** e crie 3 terminais virtuais (**CTRL+a c**):
 1. Edição
 2. Compilação, *linkedição* e testes usando o sistema operacional *Linux*.
 3. Compilação, *linkedição* e testes usando o sistema operacional *FreeBSD*.

A partir de qualquer máquina é possível se conectar remotamente a outra máquina executando o comando:

"ssh nome-da-máquina" (sem as aspas)

Para se conectar à rede DEL, usando o *MobaXterm* ou o cliente *SSH* desejado, é preciso se conectar primeiro à máquina **"loghost02.del.ufrj.br"**. A partir desta máquina é possível se conectar às máquinas dos laboratórios executando o comando:

"ssh ligXYZ" (sem as aspas)

onde **XYZ** pode algum valor nas faixas 025-044 e 073-102.

2. Crie o diretório **~/private/EEL270/2023-1/Aulas-Praticas/RCS** utilizando o comando **mkdir**. Veja na página de manual do comando como criar todos os diretórios com uma única execução do mesmo. Os nomes dos diretórios deverão ser os indicados neste roteiro, respeitando-se os caracteres maiúsculos e minúsculos e demais caracteres, como por exemplo, o hífen.
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 **aula01.h** contendo a tabela de cores **ANSI** para utilização na função **printf**. A tabela deverá ser definida utilizando-se uma macro para cada cor e para cada variante. Deverão ser criadas macros para as cores *black*, *red*, *green*, *yellow*, *blue*, *magenta*, *cyan* e *white*. Além disso, para cada cor, deverão ser criadas macros para seguintes variantes:
 - texto em negrito,
 - texto sublinhado,
 - texto com intensidade alta,
 - texto em negrito com intensidade alta,
 - cor de fundo, e
 - cor de fundo com intensidade alta.

Como exemplos, para a cor vermelha com texto regular, defina a macro **RED** com valor igual a **"\e[0;31m"**, enquanto que para a cor vermelha com texto em negrito defina a macro **RED_BOLD** com valor igual a **"\e[1;31m"**,

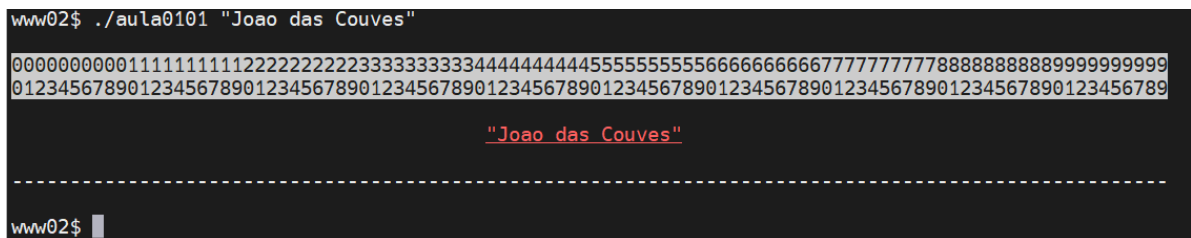
- Lembre-se que você deverá usar o padrão de nomenclatura definido em aula para os identificadores. No caso de macros deverá ser utilizado o estilo *Snake Case*. Veja observação no final do roteiro.
- O arquivo *aula01.h* deverá conter o bloco obrigatório de diretivas do preprocessador (*#ifndef*, *#define* ... *#endif*). O identificador da macro utilizada neste bloco deverá ser significativo e deverá estar relacionado ao nome do arquivo em questão. O valor desta macro deverá incluir o nome do arquivo e sua versão identificada pela macro *RCS* correspondente, ou seja, *Revision*.

```
#ifndef    _AULA_01_
#define    _AULA_01_        "@(#)aula01.h $Revision$"
...
#endif
```

5. Crie o arquivo *aula0101.c* contendo o código-fonte de um programa que receba um nome que possa conter caracteres de espaço e que tenha no máximo **64** 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 **100** colunas).

O formato de exibição deverá ser o mostrado na figura abaixo, incluindo os números das colunas:



```
www02$ ./aula0101 "Joao das Couves"
00000000001111111112222222223333333333444444444555555555666666666777777777888888888999999999
0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
"Joao das Couves"
-----
www02$
```

As duas linhas que exibem os números das colunas deverão estar com fundo branco e texto preto. O nome deverá ser exibido com texto em vermelho e sublinhado.

Note que existe uma linha em branco no início e uma linha em branco no final da saída gerada pelo programa. Além disso, existe uma linha em branco antes e após a linha contendo o nome. As linhas correspondendo à numeração das colunas, bem com a linha contendo os hifens deverão ser geradas utilizando-se laços de repetição.

A função *strlen* retorna o comprimento útil de uma *string*, ou seja, o comprimento da *string* não incluindo o caractere de final de *string* (*EOS* – *End of String*).

6. Crie os arquivos de dependências (*BSDmakefile* e *GNUmakefile*) contendo as macros *CC*, *LD*, *CFLAGS*, *LFLAGS*, *DIALECT*, *STANDARD*, *AULA01*, *AULA0101OBS*, *EXECS*, *LIBS* e *ALL*.

A macro **STANDARD** deverá receber o valor padrão **ANSI**. Este valor poderá ser alterado se durante a execução do comando **make**, o dialeto for alterado (conforme definido das observações iniciais). Neste caso, a macro deverá receber um dos seguintes valores:

-std=c89, **-std=c90**, **-std=c99** ou **-std=c11**.

As macros **AULA01** e **AULA0101OBS** deverão receber os valores **aula0101** e **aula0101.o** respectivamente. A macro **EXECS** deverá receber o valor da macro **AULA01**, enquanto que a macro **LIBS** deverá corresponder a uma *string* vazia. Finalmente, a macro **ALL** deverá corresponder aos valores das macros **EXECS** e **LIBS**.

Inclua, nos arquivos de dependências, os rótulos **all**, **aula01**, **aula0101**, **clean-all**, **clean**, **clean-objs**, **clean-bsd**, **clean-linux**, **clean-gcc**, **clean-clang**, **clean-ansi**, **clean-c89**, **clean-c90**, **clean-c99** e **clean-c11** com as declarações necessárias.

O objetivo **aula01** deverá permitir gerar todos os binários solicitados nesta aula.

Inclua, nos comandos relativos ao objetivo **aula0101**, o comando necessário para criar uma cópia do binário 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 (informações na ordem mostrada no exemplo):

aula0101-FreeBSD-gcc-ansi - versão gerada no *FreeBSD* usando o **gcc** e o padrão **ANSI**.

7. Gere e teste as 20 possíveis versões do binário **aula0101** (*Linux* x *FreeBSD*, *clang* x *gcc*, *ANSI* x *C89* x *C90* x *C99* x *C11*).
8. Submeta os arquivos **aula01.h**, **aula0101.c**, **BSDmakefile** e **GNUmakefile** ao sistema de controle de versão.
9. Recupere uma cópia de leitura dos arquivos **aula01.h** e **aula0101.c** e uma cópia de escrita dos arquivos de dependências.
10. Crie o arquivo **aula0102.c** contendo um programa que exiba os tamanhos dos 5 tipos básicos, dos 2 modificadores de sinal e dos 3 modificadores de largura.

Use o formato mostrado na figura a seguir.

Exiba os títulos correspondentes a cada bloco de informação (sem caracteres especiais):

Tamanhos dos Tipos Basicos

Tamanhos dos Modificadores de Sinal

Tamanhos dos Modificadores de Largura

Estes títulos deverão ser exibidos a partir da primeira coluna e deverão estar sublinhados. Antes do título, exiba uma linha em branco, uma linha contendo 100 hifens e uma linha em branco.

Após o título inclua uma linha em branco e a seguir os tamanhos dos tipos desejados.

O nome do tipo (exatamente igual à palavra-chave correspondente) deverá ser exibido a partir da coluna 1 usando a cor azul, enquanto que o tamanho em bytes deste tipo (utilizando-se a palavra chave **sizeof**) deverá ser exibido na mesma linha a partir da coluna 91. A palavra *byte* ou a palavra *bytes* (mantendo a concordância) deverá ser exibida após um

caractere de espaço colocado após o tamanho do tipo em questão. O tamanho e a palavra *byte/bytes* deverão ser exibido usando a cor amarela.

Após o último tipo, deverá ser incluída uma linha em branco, uma linha com 100 hifens 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*.

Operador Ternário:

Condição ? verdadeiro : falso

Onde

Condição corresponde à condição que será testada.

verdadeiro corresponde à ação que será executada quando a condição for verdadeira.

falso corresponde à ação que será executada quando a condição for falsa.

A linha contendo os 100 hifens deverá ser gerada (exibida) usando o laço de repetição desejado.

```
www02$ ./aula0102

-----

Tamanhos dos Tipos Basicos

void:                                     1 byte
char:                                    1 byte
int:                                     4 bytes
float:                                   4 bytes
double:                                 8 bytes

-----

Tamanhos dos Modificadores de Sinal

signed:                                   4 bytes
signed:                                   4 bytes

-----

Tamanhos dos Modificadores de Largura

short:                                   2 bytes
long:                                    8 bytes
long long:                              8 bytes

-----

www02$ █
```

11. Inclua, no arquivo de dependências, a macro *AULA0102OBS* e o rótulo *aula0102* com as declarações necessárias. Atualize os valores das macros, onde for necessário.
12. Gere e teste as 20 possíveis versões do executável *aula0102*.

13. Crie uma cópia do arquivo *aula0102.c* com o nome *aula0103.c*.
14. Submeta o arquivo *aula0102.c* e os arquivos de dependências ao sistema de controle de versão.
15. Recupere uma cópia de leitura do arquivo *aula0102.c* e uma cópia de escrita dos arquivos de dependências.
16. Inclua, no arquivo *aula0103.c*, as linhas necessárias para exibir os tamanhos em bytes das combinações válidas entre modificadores de sinal e modificadores de largura, entre tipos básicos e modificadores de sinal, entre tipos básicos e modificadores de largura e entre tipos básicos, modificadores de sinal e modificadores de largura.

Mantenha o formato de exibição do programa anterior, exibindo os títulos correspondentes a cada bloco de informação, ou seja:

```
Tamanhos dos Modificadores de Sinal combinados com Modificadores de Largura
```

```
Tamanhos dos Tipos Basicos combinados com Modificadores de Sinal
```

```
Tamanhos dos Tipos Basicos combinados com Modificadores de Largura
```

```
Tamanhos dos Tipos Basicos combinados com Modificadores de Sinal e com Modificadores de Largura
```

17. Inclua, no arquivo de dependências, a macro *AULA0103OBS* e o rótulo *aula0103* com as declarações necessárias. Atualize os valores das macros, onde for necessário.
18. Gere e teste as 20 possíveis versões do executável *aula0103*.
19. Submeta os arquivos *aula0103.c* e os arquivos de dependências ao sistema de controle de versão.
20. Recupere uma cópia de leitura do arquivo *aula0103.c* e uma cópia de escrita dos arquivos de dependências.
21. Limpe o diretório (*make clean-all*).
22. Arquivos que devem ser disponíveis ao final da aula:

Subdiretório "~/private/EEL270/2022-2/Aulas-Praticas"

- *aula01.h*
- *aula0101.c*
- *aula0102.c*
- *aula0103.c*
- *BSDmakefile*
- *GNUmakefile*

Subdiretório "~/private/EEL270/2022-2/Aulas-Praticas/RCS"

- *aula01,h,v*
- *aula0101.c,v*
- *aula0102.c,v*

- *aula0103.c,v*
- *BSDmakefile,v*
- *GNUmakefile,v*

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).

Estilos para Identificadores

- ***Snake Case***
 - letras maiúsculas, dígitos e o caractere sublinhado;
 - macros e constantes;
 - Exemplo: ***COMPRIMENTO_MAXIMO_NOME***
- ***Camel Case***
 - letras minúsculas com exceção do primeiro caractere de cada palavra a partir da segunda palavra que compõe o identificador;
 - tipos, membros de tipos enumerados e variáveis;
 - Exemplo: ***comprimentoNomeCompleto***
- ***Pascal Case***
 - letras minúsculas com exceção do primeiro caractere de cada palavra que compõe o identificador;
 - funções
 - Exemplo: ***CalcularFatorial***
- Todos os identificadores podem conter dígitos, mas nenhum identificador pode começar com dígito.
- Todos os identificadores devem ser significativos e sem abreviações.
- Identificadores de função devem começar com um verbo no infinitivo.