

Aula Prática 3 - Roteiro

10/08/2021 - Roteiro referente à aula prática 03 - Máximo Divisor

Comum Versões:

03/08/2021 - Versão inicial

Prazo: 12/08/2021 - 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 itens marcados com (*) deverão ser entregues até às 19:00 do dia da aula em questão.

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*, *C90*, *C99* e *C11*, enquanto que no *CentOS* deverá ser definido o argumento *dialeto* com os valores *ANSI*, *C89*, *C90*, *C99* ou *C11*. Crie uma macro, *DIALETO*, contendo o dialeto 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:

```
$RCSfile$
```

O máximo divisor comum (MDC) entre dois números inteiros não negativos X e Y pode ser definido como:

MDC (X,Y) = MDC (Y,Z) se Z (resto da divisão de X por Y) é diferente de zero. MDC (X,Y) = Y se Z é igual a zero.

MDC (X,Y) = X se Y é igual a zero.

MDC (X,Y) = Y se X é igual a zero.

MDC (X,Y) = 0 se X e Y é igual a zero (indicando condição de erro).

1. (*) Crie, baseado na definição acima, o arquivo "*aula0301.h*" contendo o protótipo da função *CalcularMaximoDivisorComum*. Esta função deverá receber dois números inteiros não negativos e deverá retornar o valor do máximo divisor comum dentre estes números. Este arquivo deverá conter também as macros e os tipos necessários para a implementação desta função.

A macro referente à combinação *ifndef* e *define* (*_AULA0301_*), deverá ser definida como uma *string* igual a:

```
"@(#)aula0301.h $Revision$"
```

O protótipo da função é definido a seguir. O tipo *ull* deverá ser definido como um sinônimo para *unsigned long long*, utilizando a palavra *typedef*, conforme descrito abaixo:

```
typedef unsigned long long ull;
```

```
ull
```

```
CalcularMaximoDivisorComum (ull, ull);
```

2. (*) Crie o arquivo "aulao301a.c" contendo a implementação da função *CalcularMaximoDivisorComum*. Esta implementação deverá utilizar recursividade.
3. (*) Crie o arquivo "aulao302.c" contendo a implementação de um programa de testes para a função *CalcularMaximoDivisorComum*. Este programa deverá receber dois números inteiros não negativos através dos argumentos de linha de comando (CLI) e deverá exibir o máximo divisor comum dentre estes dois números. Todos os tratamentos de erro necessários e que não puderam ser realizados na função *CalcularMaximoDivisorComum* deverão ser implementados neste programa.
4. (*) Inclua, nos arquivos de dependências, as macros *AULA0302AOBJS* e *AULA03*. Altere o valor da macro *EXECS*, de forma que inclua o valor da macro *AULA03*. Inclua também os objetivos *aulao3* e *aulao302a* com os comandos correspondentes.
5. (*) Gere e teste as 16 versões do executável *aulao302a*.
6. (*) Submeta os arquivos "aulao301.h", "aulao301a.c", "aulao302.c" e "*makefile" ao sistema de controle de versão.
7. (*) Recupere uma cópia de leitura dos arquivos "aulao301a.c" e "aulao302.c" e uma cópia de escrita dos arquivos "aulao301.h" e "*makefile".
8. (*) Adicione o arquivo "aulao301a.o" ao valor da macro *LIBMATEMATICARECURSAO*.
9. (*) Gere o arquivo "libmatematicarecurso.a".
10. Crie o arquivo "aulao301b.c" contendo a implementação da função *CalcularMaximoDivisorComum* utilizando o laço de repetição *do ... while*.
11. Inclua, nos arquivos de dependências, a macro *AULA0302BOBJS* e o objetivo *aulao302b* com os comandos correspondentes. Altere o valor da macro *AULA03*, incluindo o binário correspondente.
12. Gere e teste as 16 versões do executável *aulao302b*.
13. Submeta os arquivos "aulao301.h", "aulao301b.c" e "*makefile" ao sistema de controle de versão.
14. Recupere uma cópia de leitura dos arquivos "aulao301.h" e "aulao301b.c" e uma cópia de escrita dos arquivos "*makefile".
15. Adicione o arquivo "aulao301b.o" ao valor da macro *LIBMATEMATICADOWHILE*.
16. Gere o arquivo "libmatematicadowhile.a".
17. Crie o arquivo "aulao301c.c" contendo a implementação da função *CalcularMaximoDivisorComum* utilizando o laço de repetição *for*.
18. Inclua, nos arquivos de dependências, a macro *AULA0302COBJS* e o objetivo *aulao302c* com os comandos correspondentes. Altere o valor da macro *AULA03* incluindo o binário correspondente.
19. Gere e teste as 16 versões do executável *aulao302c*.
20. Submeta os arquivos "aulao301c.c" e "*makefile" ao sistema de controle de versão.
21. Recupere uma cópia de leitura do arquivo "aulao301c.c" e uma cópia de escrita dos arquivos "*makefile".
22. Adicione o arquivo "aulao301c.o" ao valor da macro *LIBMATEMATICAFOR*.
23. Gere o arquivo "libmatematicafor.a".
24. Crie o arquivo "aulao301d.c" contendo a implementação da função *CalcularMaximoDivisorComum* utilizando o laço de repetição *while*.
25. Inclua, nos arquivos de dependências, a macro *AULA0302DOBJJS* e o objetivo *aulao302d* com os comandos correspondentes. Altere o valor da macro *AULA03* incluindo o binário correspondente.
26. Gere e teste as doze versões do executável *aulao302d*.
27. Submeta o arquivo "aulao301d.c" e "*makefile" ao sistema de controle de versão.
28. Recupere uma cópia de leitura do arquivo "aulao301d.c" e uma cópia de escrita dos arquivos "*makefile".
29. Adicione o arquivo "aulao301d.o" ao valor da macro *LIBMATEMATICAWHILE*.
30. Gere o arquivo "libmatematicawhile.a".
31. Submeta os arquivos "*makefile" ao sistema de controle de versão.
32. Recupere uma cópia de escrita dos arquivos "*makefile".
33. Limpe o diretório (*make clean-all*).

Sugestão de Leitura:

1. Página de manual do comando *ar*.