eventos

novidades

página inicial | equipe | atividades | localização | contatos | área de membros | pastas das disciplinas | tutoriais | nov

Aula Prática 02 - Roteiro

20/08/2019 - Roteiro referente a aula prática 02 - Máximo Divisor Comum

Versão: 20/08/2019

Prazo: 27/08/2019 - 6:00

navegação

**Equipe** 

Atividades

Localização

Área de Membros

Computação I

Computação II

Linguagens de

🛅 EEL875 - Internet

e Arquitetura

🛅 EEL878 - Redes

🛅 EEL879 - Redes

de Computadores

de Computadores

Sistemas Digitais

Programação

Contatos

Pastas das

Disciplinas

ighthalf = 10 | EEL170 -

iii EEL270 -

EEL670 -

TCP/IP

Ι

II

Tutoriais

Webmail

Novidades

Nome do Usuário

**Eventos** 

acessar

Senha

acessar

senha?

Esqueceu sua

🛅 EEL480 -

Página Inicial

## **Observações:**

- Leia este enunciado com MUITA atenção até o final antes de iniciar o trabalho.
- Este roteiro está disponível no formato PDF. Para acessá-lo, clique aqui.
- 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 (maiúsculas, minúsculas, caracteres especiais e extensões, se existentes).
- As tarefas deverão ser executadas na ordem solicitada neste roteiro.
- A compilação e a *linkedição* deverão ser executadas utilizando-se tanto o *gcc*, quanto o *clang* . Em ambos os casos deverão ser utilizados os flags "-*Wall -std=c99*".
- Além disso, deverão ser executadas sem mensagens de advertência e sem mensagens de erro, tanto no CentOS
   7.x, quanto no FreeBSD 12.x.
- No CentOS o comando make corresponde ao GNU Make, enquanto que no FreeBSD o comando é nativo. Estas duas variantes não são cem por cento compatíveis e por isso serão necessários dois arquivos de dependências, o GNUmakefile e o BSDmakefile respectivamente. No FreeBSD o comando gmake poderia ser utilizado com o arquivo GNUmakefile, mas isto está fora do escopo desta aula.
- Inclua, no início de todos os arquivos solicitados (\*.c e \*makefile), os seguintes comentários:

Universidade Federal do Rio de Janeiro
Escola Politecnica

Departamento de Eletronica e de Computacao
EEL270 - Computacao II - Turma 2019/2
Prof. Marcelo Luiz Drumond Lanza
Autor: <nome completo>
Descricao: <descrição sucinta dos objetivos do programa>

\$Author\$ \$Date\$

\$Log\$

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. Escreva, baseado na definição acima, o arquivo "aula0201.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 destes 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*, por exemplo *AULA0201*, deverá ser definida como uma *string* igual a:

"@(#)aula0201.h \$Revision\$"

O protótipo da função é definido a seguir. O tipo *ull* deverá ser definido como um sinônimo para *unsigned long* long:

ull

 ${\tt Calcular Maximo Divisor Comum\ (ull,\ ull);}$ 

- 2. Escreva o arquivo "aula0201a.c" contendo a implementação da função CalcularMaximoDivisorComum. Esta implementação deverá utilizar recursividade.
- 3. Escreva o arquivo "aula0202.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 destes 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 *AULA0202AOBJS* e *AULA02*. Altere o valor da macro *EXECS*, de forma que inclua o valor da macro *AULA02*. Inclua também os objetivos *aula02* e *aula0202a* com os comandos correspondentes.
- 5. Crie e teste as quatro versões do executável *aula0202a*.
- 6. Faça uma cópia do arquivo "aula0201a.c" com o nome "aula0201b.c".
- 7. Submeta os arquivos "aula0201.h", "aula0201a.c", "aula0202.c" e "\*makefile" ao sistema de controle de versão.
- 8. Recupere uma cópia de leitura dos arquivos "aula0201a.c" e "aula0202.c" e uma cópia de escrita dos arquivos "aula0201.h" e "\*makefile"
- 9. Altere o arquivo "aula0201b.c" de forma que contenha a implementação da função CalcularMaximoDivisorComum utilizando o laço de repetição do ... while.
- 10. Inclua, nos arquivos de dependências, a macro *AULA0202BOBJS* e o objetivo *aula0202b* com os comandos correspondentes. Altere o valor da macro *AULA02* incluindo o binário correspondente.
- 11. Crie e teste as quatro versões do executável *aula0202b.*
- 12. Faça uma cópia do arquivo "aula0201b.c" com o nome "aula0201c.c".
- 13. Submeta os arquivos "aula0201.h", "aula0201b.c" e "\*makefile" ao sistema de controle de versão.
- 14. Recupere uma cópia de leitura do arquivo "aula0201b.c" e uma cópia de escrita dos arquivos "aula0201.h" e "\*makefile".
- 15. Altere o arquivo "aula0201c.c" de forma que contenha a implementação da função CalcularMaximoDivisorComum utilizando o laço de repetição for.
- 16. Inclua, nos arquivos de dependências, a macro *AULA0202COBJS* e o objetivo *aula0202c* com os comandos correspondentes. Altere o valor da macro *AULA02* incluindo o bináro correspondente.
- 17. Crie e teste as quatro versões do executável *aula0202c*.
- 18. Faça uma cópia do arquivo "aula0201c.c" com o nome "aula0201d.c".
- 19. Submeta os arquivos "aula0201.h", "aula0201c.c" e "\*makefile" ao sistema de controle de versão.
- 20. Recupere uma cópia de leitura do arquivo "aula0201c.c" e uma cópia de escrita dos arquivos "aula0201.h" e "\*makefile".
- 21. Altere o arquivo "aula0201d.c" de forma que contenha a implementação da função CalcularMaximoDivisorComum utilizando o laço de repetição while.
- correspondentes. Altere o valor da macro AULA02 incluindo o bináro correspondente.

22. Inclua, nos arquivos de dependências, a macro *AULA0202DOBJS* e o objetivo *aula0202d* com os comandos

- 23. Crie e teste as quatro versões do executável *aula0202d.*
- 24. Submeta os arquivos "aula0201.h", "aula0201d.c" e "\*makefile" ao sistema de controle de versão.
- "\*makefile".

25. Recupere uma cópia de leitura dos arquivos "aula0201.h" e "aula0201d.c" e uma cópia de escrita dos arquivos

buscar

acessar

Portal DEL (c) 2008 - Departamento de Engenharia Eletrônica e de Computação

PLONE POWERED

ANY BROWSER