eventos

novidades

tutoriais

localização

buscar

acessar

você está aqui: página inicial  $\rightarrow$  pastas das disciplinas  $\rightarrow$  eel270 - computação ii  $\rightarrow$  turma 2019-2  $\rightarrow$  aulas práticas  $\rightarrow$  roteiros  $\rightarrow$  aula prática 03 - roteiro

contatos

Aula Prática 03 - Roteiro navegação 27/08/2019 - Roteiro referente à aula prática 03 - Série de Fibonacci Página Inicial Versão: 27/08/2019 **Equipe Prazo:** 03/09/2019 - 6:00 Atividades Localização **Observações:** Contatos Area de Membros Pastas das Disciplinas ighthalf = 10 | EEL170 -Computação I iii EEL270 -Computação II 🛅 EEL670 -Linguagens de

atividades

equipe

página inicial

Programação

EEL875 - Internet

e Arquitetura

EEL878 - Redes

🛅 EEL879 - Redes

de Computadores

de Computadores

Sistemas Digitais

TCP/IP

Ι

II

Tutoriais

Webmail

**Eventos** 

acessar

Senha

senha?

acessar

Esqueceu sua

Novidades

Nome do Usuário

🛅 EEL480 -

 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).

área de membros

pastas das disciplinas

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 11.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. No FreeBSD o comando gmake poderia ser utilizado com o arquivo GNUmakefile, mas isto está fora do escopo desta aula.

 Inclua, sempre que necessário, o comando para criar uma cópia do binário com a identificação do sistema operacional e do compilador/linkeditor utilizados.

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

\$Author\$ \$Date\$ \$Log\$

A série de Fibonacci é dada por:

Número do Termo	0	1	2	3	4	5	6	7	8	9	
Valor do Termo	0	1	1	2	3	5	8	13	21	34	
					1						

 $F(n) = n \text{ se } n \leq 1$ F(n) = F(n-1) + F(n-2) se n > 1

1. Crie o arquivo "aula0301.h" contendo a definição do protótipo da função CalcularTermoSerieFibonacci. Esta função deverá receber um inteiro não negativo (número do termo desejado) e deverá retornar o valor deste termo. A macro referente à combinação ifndef e define, como por exemplo \_AULA0301\_, deverá ser definida como uma *string* valendo:

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

unsigned long long CalcularTermoSerieFibonacci (unsigned short);

2. Crie o arquivo "aula0301a.c" contendo o código fonte da função CalcularTermoSerieFibonacci implementada utilizando-se recursividade. A implementação desta função não poderá utilizar nenhuma função de nenhuma biblioteca.

3. Crie o arquivo "aula0302.c" contendo o código fonte de um programa de testes para a função criada na questão anterior. Este programa deverá receber, através de um argumento da CLI e utilizando a função strtoul, um inteiro não negativo representando **n** (o limite superior para a exibição dos valores da série de *Fibonacci*). O programa deverá exibir os valores dos termos da série de Fibonacci deste o elemento **0** até o elemento **n**. Se, por exemplo, n for igual a 5 a saída deverá ser:

F(0) = 0

F(1) = 1F(2) = 1

F(3) = 2F(4) = 3

F(5) = 5

4. Inclua, nos arquivos de dependências, as macros AULA03 - correspondendo ao executável aula0302a (resultado da combinação entre a função implementada utilizando-se recursividade e o programa de testes) e AULA0302AOBJS - correspondendo aos arquivos objetos necessários para gerar o executável aula0302a. Altere o valor da macro EXECS, de forma que inclua o valor da macro AULA03. Inclua também os objetivos aula03 e aula0302a.

5. Crie e teste as quatro versões do executável *aula0302a*.

6. Submeta os arquivos *aula0301.h*, *aula0301a.c*, *aula0302.c* e \**makefile* ao sistema de controle de versão.

7. Recupere uma cópia de leitura dos arquivos aula0301.h, aula0301a.c e aula0302.c e uma cópia de escrita dos arquivos \*makefile.

8. Crie o arquivo "aula0301b.c" contendo o código fonte da função CalcularTermoSerieFibonacci implementada utilizando-se a estrutura de controle "do ... while". A implementação desta função não poderá utilizar nenhuma função de nenhuma biblioteca.

9. Altere, no arquivo de dependências, a macro AULA03 - incluindo o executável aula0302b (resultado da combinação entre a função implementada utilizando-se a estrutura de controle do ... while e o programa de testes). Inclua a macro AULA0302BOBJS - correspondendo aos arquivos necessários para gerar o executável aula0302b. Inclua também o objetivo aula0302b.

10. Crie e teste as quatro versões do executável aula0302b.

11. Submeta os arquivos aula0301b.c e \*makefile ao sistema de controle de versão.

12. Recupere uma cópia de leitura do arquivo *aula0301b.c* e uma cópia de escrita dos arquivos \*makefile.

13. Crie o arquivo "aula0301c.c" contendo o código fonte da função CalcularTermoSerieFibonacci implementada utilizando-se a estrutura de controle "for". A implementação desta função não poderá utilizar nenhuma função de nenhuma biblioteca.

14. Altere, no arquivo de dependências, a macro AULA03 - incluindo o executável aula0302c (resultado da combinação entre a função implementada utilizando-se a estrutura de controle for e o programa de testes). Inclua a macro AULA0302COBJS - correspondendo aos arquivos necessários para gerar o executável aula0302c. Inclua também o objetivo *aula0302c*.

15. Crie e teste as quatro versões do executável *aula0302c*.

16. Submeta os arquivos *aula0301c.c* e \**makefile* ao sistema de controle de versão.

17. Recupere uma cópia de leitura do arquivo *aula0301c.c* e uma cópia de escrita dos arquivos \**makefile*.

18. Crie o arquivo "aula0301d.c" contendo o código fonte da função CalcularTermoSerieFibonacci implementada utilizando-se a estrutura de controle "while". A implementação desta função não poderá utilizar nenhuma função de nenhuma biblioteca.

19. Altere, no arquivo de dependências, a macro AULA03 - incluindo o executável aula0302d (resultado da combinação entre a função implementada utilizando-se a estrutura de controle while e o programa de testes). Inclua a macro AULA0302DOBJS - correspondendo aos arquivos necessários para gerar o executável aula0302d. Inclua também o objetivo aula0302d.

20. Crie e teste as quatro versões do executável *aula0302d*.

21. Submeta os arquivos aula0301d.c e \*makefile ao sistema de controle de versão.

22. Recupere uma cópia de leitura do arquivo *aula0301d.c* e uma cópia de escrita dos arquivos \*makefile.

W3C XHTML

W3C css

ANY BROWSER