

Aula Prática 02 - Roteiro

02/04/2019 - Roteiro referente à aula prática 02 - Série de Fibonacci

Versão: 02/04/2019

Prazo: 09/04/2019 - 8:00

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* 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 Computacao
EEL270 - Computacao II - Turma 2019/1
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	...

$$F(n) = n \text{ se } n \leq 1$$

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

1. Crie o arquivo `"aula0201.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 `_AULA0201_`, deverá ser definida como uma *string* valendo:

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

unsigned long long

CalcularTermoSerieFibonacci (unsigned short);

2. Crie o arquivo "aula0201a.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 "aula0202.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:

$F(0) = 0$
 $F(1) = 1$
 $F(2) = 1$
 $F(3) = 2$
 $F(4) = 3$
 $F(5) = 5$
4. Inclua, nos arquivos de dependências, as macros *AULA02* - correspondendo ao executável *aula0202a* (resultado da combinação entre a função implementada utilizando-se recursividade e o programa de testes) e *AULA0202AOBJS* - correspondendo aos arquivos objetos necessários para gerar o executável *aula0202a*. Altere o valor da macro *EXECS*, de forma que inclua o valor da macro *AULA02*. Inclua também os objetivos *aula02* e *aula0202a*.
5. Crie e teste as quatro versões do executável *aula0202a*.
6. Submeta os arquivos *aula0201.h*, *aula0201a.c*, *aula0202.c* e **makefile* ao sistema de controle de versão.
7. Recupere uma cópia de leitura dos arquivos *aula0201.h*, *aula0201a.c* e *aula0202.c* e uma cópia de escrita dos arquivos **makefile*.
8. Crie o arquivo "aula0201b.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 *AULA02* - 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 *AULA0202BOBJS* - correspondendo aos arquivos necessários para gerar o executável *aula0202b*. Inclua também o objetivo *aula0202b*.
10. Crie e teste as quatro versões do executável *aula0202b*.
11. Submeta os arquivos *aula0201b.c* e **makefile* ao sistema de controle de versão.
12. Recupere uma cópia de leitura do arquivo *aula0201b.c* e uma cópia de escrita dos arquivos **makefile*.
13. Crie o arquivo "aula0201c.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 *AULA02* - incluindo o executável *aula0202c* (resultado da combinação entre a função implementada utilizando-se a estrutura de controle for e o programa de testes). Inclua a macro *AULA0202COBJS* - correspondendo aos arquivos necessários para gerar o executável *aula0202c*. Inclua também o objetivo *aula0202c*.
15. Crie e teste as quatro versões do executável *aula0202c*.
16. Submeta os arquivos *aula0201c.c* e **makefile* ao sistema de controle de versão.
17. Recupere uma cópia de leitura do arquivo *aula0201c.c* e uma cópia de escrita dos arquivos **makefile*.
18. Crie o arquivo "aula0d01d.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 *AULA02* - incluindo o executável *aula0202d* (resultado da combinação entre a função implementada utilizando-se a estrutura de controle while e o programa de testes). Inclua a macro *AULA0202DOBJs* - correspondendo aos arquivos necessários para gerar o executável *aula0202d*. Inclua também o objetivo *aula0202d*.
20. Crie e teste as quatro versões do executável *aula0202d*.
21. Submeta os arquivos *aula0201d.c* e **makefile* ao sistema de controle de versão.
22. Recupere uma cópia de leitura do arquivo *aula0201d.c* e uma cópia de escrita dos arquivos **makefile*.