eventos

buscar

acessar

novidades

você está aqui: página inicial o pastas das disciplinas o eel270 - computação ii o turma 2019-2 o aulas práticas o roteiros o aula prática 10 - roteiro

contatos

página inicial equipe navegação Página Inicial **Equipe** Atividades Localização Contatos Area de Membros Pastas das Disciplinas EEL170 -Computação I iii EEL270 -Computação II EEL670 -Linguagens de Programação EEL875 - Internet e Arquitetura TCP/IP 🛅 EEL878 - Redes de Computadores Ι 🛅 EEL879 - Redes de Computadores IIEEL480 -Sistemas Digitais



Esqueceu sua

senha?

Aula Prática 10 - Roteiro

localização

22/10/2019 - Roteiro referente à aula prática 10 - Arquivos texto formato Unix versus arquivos texto formato Microsoft. Funções getopt, getopt_log e getsubopt.

área de membros

pastas das disciplinas

tutoriais

Versão: 22/10/2019

Prazo: 05/11/2019 - 8:00

Observações:

atividades

- 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 2019/2
Prof. Marcelo Luiz Drumond Lanza

Descricao: <descrição sucinta dos objetivos do programa>

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

Autor: <nome completo>

- Inclua, nos arquivos de dependência, as macros LIBBASE64OBJS (correspondendo ao arquivo "aula0901.0") e LIBBASE64 (correspondendo ao arquivo "libbase64.a"). Altere o valor da macro LIBS de forma que inclua a biblioteca libbase64. Inclua também o objetivo "libbase64.a". Este obejtivo deverá ter como dependência o valor da macro LIBBASE64OBJS.
- 2. Crie o arquivo "aula1001.h" contendo a definição do tipo tipoErros e os protótipos da funções ConverterArquivoFormatoUnixFormatoDos e ConverterArquivoFormatoDosFormatoUnix. Estas funções deverão receber dois argumentos do tipo string. O primeiro contendo o nome do arquivo original e o segundo o nome do arquivo que será gerado pela conversão. As funções deverão retornar ok ou o código de erro apropriado.

tipoErros

ConverterArquivoFormatoUnixFormatoDos(char *original, char *convertido);

tipoErros Converte

ConverterArquivoFormatoDosFormatoUnix (char *original, char *convertido);

- 3. Crie o arquivo "aula1001.c" contendo a implementação das funções solicitadas na questão anterior. Lembre-se que o final de linha no formato *Unix* é composto por '\n', enquanto que o final de linha no formato *Microsoft* (DOS) é composto por '\r' e '\n'. A implementação destas funções só poderá utilizar as funções de manipulação de arquivos texto apresentadas na sala de aula, ou seja, *fopen*, *fclose*, *fgets*, *fprintf*, *feof* e *ferror*.
- 4. Crie o arquivo "aula1002.c" contendo a implementação de um programa que receba, via argumentos da interface de linha de comando, a opção curta desejada (dentre as permitidas pelo programa). As opções curtas aceitas deverão ser:
 - m | M converter um arquivo texto do formato *Unix* para o formato *Microsoft* (DOS).
 - u | U converter um arquivo texto do formato *Microsoft* para o formato *Unix*. e | E - codificar o arquivo usando o algoritmo Base64.
 - d | D decodificar o arquivo que foi previsamente codificado com o algoritmo Base64. h | H - exibir uma mensagem contendo as informações sobre o uso do programa.

Considere que todas as opções curtas NÃO possuem argumento obrigatório. Nos casos das opções d, D, e, E, m, M, u e U, o programa deverá receber via argumentos de linha de comando (use a variável optind) os nomes de dois arquivos.

- 5. Inclua, nos arquivos de dependências, a macro AULA1002OBJS correspondendo aos arquivos necessários para criar o executável a partir dos arquivos "aula1001.c" e "aula1002.c". Além disso, crie a macro AULA10 correspondendo ao executável "aula1002". Os arquivos de dependências deverão incluir ainda o objetivo aula1002 (executável criado a partir dos arquivos definidos pela macro AULA1002OBJS.
- 6. Crie e teste as quatro versões do executável *aula1002*.
- 7. Submeta os arquivos "aula1001.h", "aula1001.c", "aula1002.c" e "*makefile" ao sistema de controle de versão.
- 8. Recupere uma cópia de leitura dos arquivos "aula1001.h", "aula1001.c" e "aula1002.c" e uma cópia de escrita dos arquivos "*makefile".
- 9. Crie o arquivo "aula1003.c" contendo a implementação de um programa que receba, via argumentos da interface de linha de comando, a opção curta ou a opção longa desejada (dentre as permitidas pelo programa). As opções curtas aceitas deverão ser:
 - m | M | dos converter um arquivo texto do formato *Unix* para o formato *Microsoft* (DOS).
 - u | U | unix converter um arquivo texto do formato *Microsoft* para o formato *Unix*. e | E | encode - codificar o arquivo usando o algoritmo Base64.
 - d | D | decode decodificar o arquivo que foi previsamente codificado com o algoritmo Base64.
 h | H | help exibir uma mensagem contendo as informações sobre o uso do programa.

Considere que todas as opções NÃO possuem argumento obrigatório. Nos casos das opções d, D, e, E, m, M, u e U (e opções longas correspondentes), o programa deverá receber via argumentos de linha de comando (use a

- 10. Inclua, nos arquivos de dependências, a macro AULA1003OBJS correspondendo aos arquivos necessários para criar o executável a partir dos arquivos "aula1001.c" e "aula1003.c". Além disso, altere a macro AULA10 incluindo o executável "aula1003". Os arquivos de dependências deverão incluir ainda o objetivo aula1003 (executável criado a partir dos arquivos definidos pela macro AULA1003OBJS.
- 11. Crie e teste as quatro versões do executável aula1003.

variável optind) os nomes de dois arquivos.

- 12. Submeta os arquivos "aula1003.c" e "*makefile" ao sistema de controle de versão.
- 13. Recupere uma cópia de leitura do arquivo "aula1003.c" e uma cópia de escrita dos arquivos "*makefile".
- 14. Crie o arquivo "aula1004.c" contendo a implementação de um programa que receba, via argumentos da interface de linha de comando, a opção curta ou a opção longa desejada (dentre as permitidas pelo programa). As opções curtas aceitas deverão ser:
 - m | M | dos converter um arquivo texto do formato *Unix* para o formato *Microsoft* (DOS).
 u | U | unix converter um arquivo texto do formato *Microsoft* para o formato *Unix*.
 - u | U | unix converter um arquivo texto do formato *Microsoft* para o formato *Unix*. e | E | encode - codificar o arquivo usando o algoritmo Base64.
 - d | D | decode decodificar o arquivo que foi previsamente codificado com o algoritmo Base64.
 - h | H | help exibir uma mensagem, no idioma selecionado, contendo as informações sobre o uso do programa.

Considere que todas as opções NÃO possuem argumento obrigatório. Os argumentos necessários para cada opção deverão ser obtidos usando a função getsubopt. As palavras-chave utilizadas deverão ser *input*, *output* e *language*, correspondendo respectivamente ao nome do arquivo de entrada, ao nome do arquivo de saída e ao idioma utilizado.

criar o executável a partir dos arquivos "aula1001.c" e "aula1004.c". Além disso, altere a macro AULA10 incluindo o executável "aula1004". Os arquivos de dependências deverão incluir ainda o objetivo aula1004 (executável criado a partir dos arquivos definidos pela macro AULA1004OBJS.

15. Inclua, nos arquivos de dependências, a macro AULA1004OBJS - correspondendo aos arquivos necessários para

- 16. Crie e teste as quatro versões do executável *aula1004.*
- 17. Submeta os arquivos "aula1004.c" e "*makefile" ao sistema de controle de versão.
- 18. Recupere uma cópia de leitura do arquivo "aula1004.c" e uma cópia de escrita dos arquivos "*makefile".

Supervisionado -

05/03/2012

Mais notícias...

Importante

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

ANY BROWSER