Aula Prática 10 - Roteiro

05/10/2021 - Roteiro referente à aula prática 10 - Implementação de CGI´s.

Versão: 01/10/2021

Prazo: 18/10/2021 - 18:00

Valor: 10,0 - Peso: 3

- Leia este enunciado com **MUITA** atenção até o final antes de iniciar o trabalho.
- 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.**
- 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 definida uma macro (no *FreeBSD*) ou um argumento com o valor desejado (no *CentOS*). As duas macros 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, C99 e C11, enquanto que no *CentOS* deverá ser definido o argumento *dialeto* com os valores ANSI, 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 "c99" ou "c11" de acordo com o esquema definido acima.
- O *flag* de linkedição deverá ser igual a "-Wall" (sem as aspas).
- 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 -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:

```
$RCSfile$
```

1. Copie os arquivos "*teste-centos.cgi*" e "*teste-freebsd.cgi*" disponíveis no diretório "*~marcelo.lanza/public/html*" para o seu diretório "*~/public/html*".

2. Teste estas duas CGIs nos servidores *web* correspondentes. Caso ocorra algum erro, através do grupo da disciplina no Whatsapp, ao professor para que as correções necessárias sejam realizadas na sua conta.

3. Inclua, nos arquivos de dependências, a macro ***AULA10*** e o objetivo ***aula10***. A macro deverá corresponder a todos os executáveis solicitados neste roteiro, tanto os executáveis da CLI, quanto as CGIs. O roteiro ***aula10*** deverá ter como dependência o valor da macro ***AULA10***.

   Atualize os arquivos de dependências sempre que for necessário.

4. Crie, no diretório correspondente, ou seja, "*/users/**username**/private/EEL270/2020-2/Aulas-Praticas*", os arquivos "*indice_pt-br.html*" e "*indice_en-us.html*". Estes arquivos deverão conter os códigos HTML necessários para implementar o redirecionamento imediato para a CGI (*Common Gateway Interface*) "*eelExibirPaginaInicial.cgi*" incluindo o argumento "*eelIdioma* com o valor correspondente (*eelPortugues* e *eelIngles* respectivamente).

   Dicas:

   - Veja a meta informação *REFRESH* (HTML) - www.w3schools.com

   - Argumentos podem ser passados para uma CGI na própria URL (*Uniform Resource Locator*). Como exemplo, para uma CGI cujo nome é "fatorial.cgi", que tem um argumento cujo nome é "numero" recebendo o valor "10", a URL correspondente seria <a href="fatorial.cgi?numero=10">, supondo-se que a CGI está armazenada no mesmo diretório em que está armazenado o código da página em questão.

   - Para passar dois ou mais argumentos para a CGI o caractere & deverá ser utilizado entre o valor de um argumento e o nome do próximo argumento.

   - No caso deste trabalho, a URL deverá incluir também o diretório no qual a CGI está armazenada:

   ...="./CGIs/eelExibirPaginaInicial.cgi?eelIdioma=eelPortugues".

5. Cire o arquivo "*eelTipos.h*" contendo a definição do tipo enumerado *eelTipoIdiomas*, incluindo os elementos *eelPortugues* e *eelIngles*.
6. Crie o arquivo "*eelErros.h*" contendo a definição do tipo enumerado *eelTipoErros* (códigos de erro/retorno) e dos protótipos das funções *EelObterMensagemErroCli* e EelObterMensagemErroWeb.

   *char **
   EelObterMensagemErroCli (eelTipoIdiomas, eelTipoErros);

   *char **
   EelObterMensagemErroWeb (eelTipoIdiomas, eelTipoErros);

7. Crie o arquivo "*eelErros.c*" contendo a definição das variáveis "globais" *eelMensagensErroCli* e *eelMensagensErroWeb*. Este arquivo deverá conter também as implementações das funções

EelObterMensagemErroCli e *EelObterMensagemErroWeb*.

Os valores das variáveis acima deverão ser atualizados sempre que for necessário definir um novo código de erro.

Exemplo:

```
char *eelMensagensErro [eelQuantidadeIdiomas] [eelQuantidadeErros] =
{
  {
    "Sucesso",
    "Erro relacionado com arquivo",
    "Erro alocando memória",
    "Senha incorreta"
  },
  {
    "Ok",
    "File error",
    "Memory alocation error",
    "Invalid password"
  }
};
```

8. Crie o arquivo "eelInterfaceUsuario.h" contendo a definição do tipo enumerado *eelTipoNumerosMensagemInterfaceUsuario* e dos protótipos das funções *EelObterMensagemInterfaceUsuarioCli* e *EelObterMensagemInterfaceUsuarioWeb*.

   ```
   char *
   EelObterMensagemInterfaceUsuarioCli (eelTipoIdiomas,
   eelTipoNumerosMensagemInterfaceUsuario );

   char *
   EelObterMensagemInterfaceUsuarioWeb (eelTipoIdiomas,
   eelTipoNumerosMensagemInterfaceUsuario );
   ```

9. Crie o arquivo "*eelInterfaceUsuario.c*" contendo a definição das variáveis "globais" *eelMensagensInterfaceUsuarioCli* e *eelMensagensInterfaceUsuarioWeb*. Estas variáveis deverão ser definidas de forma semelhante àquela utilizada nas definições das variáveis *eelMensagensErroCli* e *eelMensagensErroWeb* e deverão conter as *strings* necessárias para gerar a interface com o usuário nos dois idiomas, tanto na linha de comando, quanto via web.

   Como um exemplo, no caso da CLI uma possível *string* seria "*Usuario*" e "*Username*", enquanto que via web seria "*Usuário*" e "*Username*".

   Este arquivo deverá conter também as implementações das funções *EelObterMensagemInterfaceUsuarioCli* e *EelObterMensagemInterfaceUsuarioWeb* .

10. Crie o arquivo "*eelCgiExibirPaginaInicial.c*" contendo o código-fonte referente à CGI "*eelExibirPaginaInicial.cgi*". Esta CGI deverá receber o idioma desejado (via argumento *eelIdioma*) e deverá gerar dinamicamente o código HTML referente à página principal do sistema. O layout desta página é livre, mas deverá incluir:

    a) o nome do sistema (Computação II - Turma 2020/2);

    b) um formulário com:

    - campo do tipo texto correspondendo ao *nickname* do usuário (nome do campo igual a *eelUsuario*), no formato nome.sobrenome - idêntico ao *nickname* usado na rede DEL, ou seja, composto por no mínimo 3 e por no máximo 65 caracteres, sendo permitidas apenas letras minúsculas e um e somente um ponto;

- campo do tipo senha correspondendo à senha do usuário (nome do campo igual a *eelSenha*), composto por no mínimo 8 e por no máximo 127 caracteres (verifique na Internet qual o conjunto de caracteres válidos para a definição de senhas no FreeBSD/Linux);

- um campo do tipo escondido contendo o idioma utilizado para criar a página (nome do campo igual a *eelIdioma*). O valor deste campo deverá ser igual a *eelPortugues* ou *eelIngles*;

- um campo do tipo submeter (botão) para executar a CGI correspondente, ou seja, *eelAutenticarUsuario.cgi*".

c) um atalho para a própria CGI ("*eelExibirPaginaInicial.cgi*") que permita mudar para o outro idioma;

d) um atalho para a CGI "*eelExibirFormularioAtivacaoContaUsuario.cgi*".

e) um atalho para a CGI "*eelExibirFormulárioReinicializacaoSenhaUsuario.cgi*".

Estes atalhos (URLs) deverão incluir o argumento *eelIdioma* com o valor correspondente ao idioma em questão.

Para implementar as CGIs crie, no diretório AulAs-Praticas, os subdiretórios Bibliotecas, Bibliotecas/FreeBSD e Bibliotecas/CentOS. Copie os arquivos "*mlcgi.h*" e "*sendmail.h*" do diretório "/users/marcelo.lanza/public/EEL270/2020-2/Aulas-Teoricas/Bibliotecas" para o diretório Bibliotecas. Copie as versões CentOS dos arquivos "libmlcgi.a" e "libsendmail.a" do diretório "/users/marcelo.lanza/public/EEL270/2020-2/Aulas-Teoricas/Bibliotecas/CentOS" para o diretório Aulas-Praticas/CentOS e as versões FreeBSD destes arquivos do diretório /users /marcelo.lanza/public/EEL270/2020-2/Aulas-Teoricas/Bibliotecas/FreeBSD" para o diretório Aulas-Praticas/FreeBSD.

Para a implementação das CGIs solicitadas neste roteiro, defina as macros definindo comprimentos mínimo e máximo, caracteres válidos, etc. e os tipos que julgar necessários nos arquivos "eelConst.h″ e "eelTipos.h″ respectivamente.

É permitido e desejável que funções auxiliares sejam implementadas para facilitar a implementação das CGIs. Os protótipos e as implementações destas funções deverão ser incluídos nos arquivos "eelFuncoes.h" e "eelFuncoes.c" respectivamente.

As boas práticas de programação definidas durante as aulas teóricas, incluindo os padrões de nomenclatura para identificadores de macros, tipos, variáveis e funções, deverão ser utilizadas.

11. Crie o arquivo "*eelAdicionarUsuarios.c*" contendo o código-fonte de um programa que possa ser executado apenas via CLI e que permita adicionar as informações sobre um novo usuário no arquivo "*usuarios*". A cada execução do programa deverão ser incluídas as informações de um único usuário. Na primeira execução do programa o arquivo "*usuarios*" deverá ser criado. O programa deverá receber todos os valores necessários (com exceção da senha e da confirmação da senha) via argumentos de linha de comando.

    O arquivo usuários (do tipo texto) deverá ser criado e armazenado do diretório *Dados* (que deverá ser criado usando o comando *mkdir*, como subdiretório do *Aulas-Praticas*).

    Cada linha do arquivo "usuarios" deverá conter os dados de um único usuário, ou seja:

    - *nickname* do usuário;
    - senha (cifrada utilizando o algoritmo SHA256 - para os usuários com a conta inativa - ou o algoritmo SHA512 - para usuários com a conta ativa);
    - nome completo (comprimentos mínmo e máximo iguais a 5 e 100 respectivamente);
    - endereço eletrônico (verifique na Internet quais as especificações de endereços eletrônicos, conjunto de caracteres válidos, comprimentos mínimo e máximo, etc.).

    Os campos referentes a um usuário deverão ser separados pelo caractere dois pontos. Além disso, o arquivo deverá estar ordenado (ordem crescente), usando o campo *nickname* para a ordenação.

Linhas começando com o caractere # representam comentários.

As senhas deverão ser cifradas sempre com o maior *salt* permitido para o algoritmo em questão.

Para a implementação deste programa podem e devem ser criadas as funções auxiliares.

12. Execute o programa *eelAdicionarUsuarios* algumas vezes criando o arquivo "usuarios" de forma que o mesmo contenha as informações sobre pelo menos 10 usuários diferentes. Todos estes usuários deverão estar com as respectivas contas desativadas.

13. Crie o arquivo "*eelCgiExibirFormularioAtivacaoContaUsuario.c*" contendo o código fonte referente à CGI "*eelExibirFormularioAtivacaoContaUsuario.cgi*". Esta CGI deverá criar, no idioma definido pelo argumento *eelIdioma*, o código HTML referente à página de solicitação de ativação da conta de um usuário. Esta página deverá conter o nome do sistema e um formulário com os campos *eelUsuario* (tipo texto), *eelSenhaAtual* , *eelNovaSenha*, *eelConfirmacaoNovaSenha* (estes três últimos do tipo senha), *eelIdioma* (tipo escondido) e um botão para execução da CGI correspondente, ou seja, "*eelAtivarContaUsuario.cgi*".

14. Crie o arquivo "*eelCgiAtivarContaUsuario.c*" contendo o código fonte referente à CGI "*eelAtivarContaUsuario.cgi*". Esta CGI deverá receber os valores dos argumentos e*elIdioma*, *eelUsuario*, *eelSenhaAtual*, *eelNovaSenha* e *eelConfirmacaoNovaSenha*.

    A CGI *eelAtivarContaUsuario.cgi* deverá verificar se a senha do usuário em questão está armazenada no formato SHA256. Se a senha atual (armazenada no arquivo "usuarios") estiver neste formato, esta CGI deverá verificar se a senha atual fornecida pelo usuário corresponde à senha atual armazenada no arquivo. Em caso afirmativo, o próximo passo será verificar se os valores correspondentes à nova senha e à confirmação da nova senha são idênticos. Neste caso, a nova senha deverá ser cifrada utilizando o algoritmo SHA512 e armazenada no arquivo "usuarios", substituindo a senha anterior (cifrada utilizando o algoritmo SHA256). Em todos os demais casos, a página de erro correspondente deverá ser exibida.

15. Crie o arquivo "*eelCgiExibirFormularioReinicializacaoSenhaUsuario.c*" contendo o código fonte referente à CGI "*eelExibirFormularioReinicializacaoSenhaUsuario.cgi*". Esta CGI deverá criar, no idioma definido pelo argumento *eelIdioma*, o código HTML referente à página de solicitação de reinicialização da senha de um usuário. Esta página deverá conter no mínimo o nome do sistema e um formulário com os campos *eelUsuario* e *eelEmail* (ambos do tipo texto), *eelIdioma* (tipo escondido) e um botão para execução da CGI correspondente, ou seja, "*eelReinicializarSenhaUsuario.cgi*".

16. Crie o arquivo "*eelCgiReinicializarSenhaUsuario.c*" contendo o código fonte , referente à CGI "*eelReinicializarSenhaUsuario.cgi*". Esta CGI deverá receber os valores dos argumentos *eelIdioma*, *eelUsuario* e *eelEmail*. Esta CGI deverá verificar se o endereço eletrônico corresponde ao endereço do usuário em questão, ou seja, se corresponde ao endereço eletrônico cadastrado no arquivo "usuarios". Se corresponder, deverá gerar uma senha aleatória (16 caracteres alfanuméricos) que deverá ser enviada para o endereço eletrônico do usuário e deverá ser armazenada em um arquivo binário (cujo nome será igual ao *nickname* do usuário). Neste arquivo, o primeiro campo deverá ser a expiração absoluta deste procedimento (a expiração relativa deste procedimento deverá ser igual a 24 horas), seguido pela senha temporária deste usuário.

    No email enviado para o usuário em questão, deverá ser enviada também a URL correspondente à CGI que gera o formulário necessário para confirmar a reinicialização da senha do usuário, ou seja, "*eelExibirFormularioConfirmacaoReinicializacaoSenhaUsuario*.cgi". Nesta URL deverão ser incluídos os argumentos *eelIdioma* e *eelUsuario*.

17. Crie o arquivo "*eelCgiExibirFormularioConfirmacaoReinicializacaoSenhaUsuario.c*" contendo o código fonte referente à CGI "*eelExibirFormularioConfirmacaoReinicializacaoSenhaUsuario.cgi*". Esta CGI deverá criar, no idioma definido pelo argumento *eelIdioma*, o código HTML referente à página de solicitação de confirmação da reinicialização da senha de um usuário. Esta página deverá conter no mínimo o

nome do sistema e um formulário com os campos *eelUsuario* (tipo texto) e *eelSenha* (tipo password), *eelIdioma* (tipo escondido) e um botão para execução da CGI correspondente, ou seja, "*eelConfirmarReinicializacaoSenhaUsuario.cgi*".

18. Crie o arquivo "*eelCgiConfirmarReinicializacaoSenhaUsuario*.c" contendo o código fonte referente à CGI "*eelConfirmarReinicializacaoSenhaUsuario.cgi*". Esta CGI deverá receber os valores dos argumentos *eelIdioma*, *eelUsuario*, *eelSenhaAtual* (provisória e armazenada no arquivo binário identificado pelo nickname do usuário em questão), *eelNovaSenha* e *eelConfirmacsaoNovaSenha*. A CGI deverá verificar se o arquivo binário em questão existe. Se o arquivo existir a validade do procedimento de reinicialização da senha deverá ser validada. Em primeiro lugar, deverá ser verificado se o procedimento ainda não expirou. A seguir, a senha atual recebida deverá ser comparada com a senha armazenada no arquivo em questão. Se forem idênticas, a nova senha e a confirmação da nova senha deverão ser comparadas. Se forem idênticas, o arquivo em questão deverá ser apagado e a nova senha, após ser cifrada utilizando o algoritmo SHA512, deverá ser armazenada na linha correspondente ao usuário (no arquivo "*usuarios*"). Nos demais casos a página de erro correspondente deverá ser gerada.

19. Crie o arquivo "*eelCgiAutenticarUsuario.c*" contendo o código-fonte referente à CGI "*eelAutenticarUsuario.cgi*". Esta CGI deverá receber os valores dos argumentos *eelIdioma, eelUsuario e eelSenha*. Após verificar os valores dos campos correspondentes ao usuário e à senha, se válidos, a CGI deverá gerar um *cookie* cujo nome deverá ser igual ao valor do argumento *eelUsuario*, cujo valor deverá ser igual a uma *string* com comprimento igual a 1024 caracteres gerados de forma aleatória a partir do conjunto *Base64* e cuja validade deverá igual a 10 minutos). Este *cookie* deverá ser enviado para o navegador e deverá ser armazenado em um arquivo binário (cujo nome será igual ao nome do *cookie* com a extensão *cookie*) contendo a expiração absoluta e valor do *cookie* e o endereço IPv4 do navegador. Por fim, deverá gerar o código HTML de uma página contendo as URLs para as CGIs "*eelObterNotasAluno.cgi*" e "*eelLogout.cgi*".

20. Crie o arquivo "*eelAdicionarNotasAluno.c*" contendo o código-fonte de um programa que possa ser executado para criar o arquivo contendo as 10 notas de um aluno. O nome do arquivo deverá ser igual ao *username* do aluno em questão, seguid pela extensão "notas", por exemplo, se o username do aluno é "joao.silva", o nome do arquivo deverá ser "joão.silva.notas". O programa deverá receber todos os valores necessários via argumentos de linha de comando. Após verificar se o aluno em questão existe, o arquivo correspondente deverá ser criado, contendo as 10 notas do alunos em questão. O arquivo deverá ser binário. Cada nota poderá varia de "0.0" a "10.0".

21. Execute o programa *eelAdicionarNotasAluno* algumas vezes e crie os arquivos de notas de alguns alunos. Não crie este arquivo para cada aluno cadastrado. É importante ter alunos sem notas para a realização dos testes.

22. Crie o arquivo "*eelCgiObterNotasAluno.c*" contendo o código-fonte referente à CGI "eelObterNotasAluno.cgi". Esta CGI deverá receber os valores dos argumentos *eelIdioma* e *eelUsuario*. Após obter o *cookie* retornado pelo navegador deverá verificar se o mesmo é válido (utilizando para isso os dados armazenados no arquivo de *cookie* correspondente). Se o *cookie* for válido, as notas do aluno em questão deverão ser obtidas a partir do arquivo de notas referente ao aluno em questão.

23. Crie o arquivo "*eelCgiLogout.c*" contendo o código-fonte referente à CGI "*eelLogout.cgi*".

24. Inclua os rótulos *install* e *deinstall* nos arquivos de dependências. A partir do rótulo install os arquivos *.html e *.cgi deverão ser copiados para o diretório ~/public/html/eel. Este diretório também deverá ser criado a partir deste rótulo. Os comandos correspondentes ao rótulo install deverão incluir a criação de um *link* simbólico denominado "index.html" que aponte para o arquivo "indice_pt-br.html". O rótulo *deinstall* deverá ser utilizado para apagar o diretório ~public/html/eel e o seu conteúdo.