Aprendendo C com Jogo da Forca

Trabalho 2

Informações

- Trabalho **em trios**;
- Entregar em <`http://trab.dc.unifil.br/moodle/`>

1 Instruções

Com trabalho anterior atingimos o objetivo de conhecer e adquirir certa fluência e habilidade com a Linguagem C. Neste trabalho, vamos aprofundar o nosso entendimento sobre aspectos importantes da linguagem, como apontadores, *structs* e arquivos cabeçalho (aqueles terminados em `.h`). São esses recursos que diferenciam C das outras linguagens de programação, e a torna especialmente útil para programação de Sistemas Operacionais e módulos relacionados.

Além disso, qualquer linguagem de programação precisa de uma biblioteca padrão para possibilitar operações de *E/S* (Entrada e Saída), e oferecer outras funcionalidades úteis, que facilitam a vida do programador. O aprendizado de algumas das principais rotinas da biblioteca padrão de C também é tema deste trabalho.

Como sempre, boas referências da Internet e ao menos um livro sobre a linguagem são essenciais para ter sucesso.

1.1 Ferramentas

Todos as atividades desta lista deverão ser feitos com as seguintes ferramentas:

- Programados na *Linguagem C*;
- Editados com *vim*, *emacs* ou *Gedit*;
- Compilados com `gcc` ou `clang`;
- Projeto gerenciado com `make` e *Makefile*;

Para entrega do trabalho, empacotar e compactar todos os arquivos fontes e o *Makefile*, excluindo binários e outros desnecessários para compilação. Só serão aceitos trabalhos entregues no formato `.tar.gz`, utilizando a ferramenta `tar` do terminal *bash*.

Além disso, o trabalho deverá ser compilado corretamente com uma simples chamada ao `make` no terminal bash, no diretório do projeto. Essa condição é **fundamental** para o trabalho ser corrigido!

1.2 Compilação

Além de ter que ser obrigatoriamente compilado corretamente como apenas uma chamada simples ao `make` do terminal, este trabalho também impõe as seguintes funcionalidades no estagio de compilação:

- A compilação de cada objeto (`.c` para `.o`) e a link-edição (união de todos os `.o` em um binário executável) deverão ser feitas com as opções `-Wall -Werror`, como no exemplo:

```
$ clang -Wall -Werror -c hello.c
$ clang -Wall -Werror -o hello hello.o
```

- Supondo um projeto com estágio de compilação com múltiplos objetos, caso um código fonte seja alterado, ao invocar o *make*, somente os objetos dependentes desse arquivo modificado deverão ser compilados.

O não cumprimento de qualquer dessas regras resultará em corte de metade dos pontos do trabalho.

2 Roteiro de Trabalho

1. Baixe o projeto `forca.tar.gz` no Portal de Entregas da Computação e expanda-o em um diretório de trabalho qualquer a sua escolha. A sua primeira tarefa é escrever um Makefile que compile corretamente o programa. Segue algumas dicas:

   - Cada arquivo `.c` dá origem a um objeto `.o`;
   - Verifique em cada `.c` quais arquivos cabeçalho `.h` eles utilizam. Eles são dependências do objeto que o `.c` gerará;
   - O binário executável final é gerado pela união (link-edição) de todos os objetos `.o`. Logo, cada objeto desse é uma dependência do programa principal;
   - Procure utilizar variáveis como `NAME`, `OBJECTS`, `CC` e `CFLAGS`;
   - Não esqueça da regra `clear`, que apaga somente os objetos e o binário. Utilize essa regra antes de comprimir o trabalho para entrega.

   Caso consiga compilar o projeto adequadamente, você conseguirá jogar algumas partidas de Forca com palavras contidas no arquivo `palavras.txt`, ou qualquer outro que você passar como parâmetro de linha de comando.

2. Este programa utiliza várias funções da biblioteca padrão da Linguagem C, como `strlen`, `memset` e `scanf`.

   (a) Vasculhe todo o código-fonte do programa a procura de funções que o próprio programa não declara nem define (ou seja, que ele próprio não implementa), e anote-os, um em cada linha, no arquivo `c_ansi.txt`.

(b) Procure em livros ou em páginas da Internet sobre o que cada uma dessas funções fazem e faça um resumo sobre sua funcionalidade. O resumo não pode ter mais que 200 caracteres. Anote o resumo à frente do nome da função no arquivo `c_ansi.txt`, como no exemplo:

```
strlen(const char * str): retorna o comprimento da string str
```

3. A próxima tarefa será entender o código do programa. Infelizmente, o programador descuidado desse jogo de forca esqueceu de comentar o código, tornando a compreensão mais trabalhosa. Por tanto, o seu trabalho é entender e colocar um comentário pertinente para cada bloco contíguo de código, que estão separados por linhas em branco. Lembre-se também de comentar a funcionalidade de cada função, ao estilo *Javadocs*[1].

4. A atual implementação desse jogo de Forca não aceita palavras compostas, como por exemplo "*Atlético Mineiro*". Estenda a funcionalidade do programa para que esse tipo de palavra seja possível.

[1] C e C++ possuem um *Javadocs* próprio, chamado de *Doxygen*. Se preferir, pode comentar neste formato.