**Instituto Superior de Engenharia de Lisboa**

**Semestre de Verão 2013/2014**



**PROGRAMAÇÃO DE SISTEMAS COMPUTACIONAIS**

**Relatório 3ªSérie**

**Grupo 1:**

**Ana Sequeira – 35479**

**Diogo Poeira – 36238**

**Daniel Silva – 39119**

Turma LI31D

**Docente**:

Carlos Martins

Índice

[Introdução 3](#_Toc389389333)

[Exercício 3 – Biblioteca dinâmica e *makefile* 5](#_Toc389389334)

[Conclusão 6](#_Toc389389335)

[Bibliografia 7](#_Toc389389336)

## Exercício 3 – Biblioteca dinâmica e *makefile*

Na terceira alínea é pedido que se crie uma biblioteca dinâmica. Para tal, e visto ser-nos pedido no enunciado que se crie um *makefile* para gerar executáveis e bibliotecas, foi necessário criar um *makefile* para criar a biblioteca, assim como o executável para o quarto exercício. Após visualização e análise de vários exemplos diferentes de *makefile*, percebemos o seguinte:

- É boa prática criar “*labels”* com todas as flags que vão ser utilizadas pelo compilador (neste caso gcc);

- A estrutura é bastante simples, sendo que é apenas necessário perceber que o que está do lado esquerdo é o que vamos gerar e o que está do lado direito é o necessário para o gerar (a criação do que está do lado esquerdo depende de tudo o que está do lado direito), e que depois dessa definição, é necessário chamar o compilador e dizer-lhe como compilar e o que compilar;

- Ao correr no terminal o comando *make*,este apenas vai gerar a primeira label do ficheiro *makefile*, assim sendo, criámos uma *label* que irá gerar a biblioteca e o executável do exercício 4;

- É também boa prática ter uma label que remova os ficheiros, que no nosso caso, irá remover todos os ficheiros com a extensão .o, a biblioteca , os ficheiros executáveis e ainda a pasta criada pelo executável do exercício 4;

Uma vez que o makefile executa comandos da linha de comandos, para a remoção da pasta gerada pelo exercício 4, foi utilizado o comando rm , tal como para a remoção de todos os outros ficheiros. Ao tentar executar o clean , deparámo-nos com um erro ,” rm: cannot remove `Thoth': Is a directory “. Após uma rápida pesquisa, percebemos que é necessária mais uma flag : -r. Esta flag faz com que sejam removidos os ficheiros em forma hierárquica, isto é, remove todos os ficheiros que estejam dentro da pasta e só depois é que remove a pasta.

Para a geração da biblioteca (extensão .so), é necessário gerar todos os ficheiros com extensão .o dos quais a biblioteca vai depender. Para além dessa necessário, é ainda preciso criar um ficheiro do tipo .h com a assinatura de todos os métodos e todas as bibliotecas desejadas. No nosso caso, decidimos meter todos os #include que são comuns em todos os módulos na nossa biblioteca, sendo que assim apenas é feito um #include (à excepção de dois módulos que utilizam ainda outra biblioteca)

## Bibliografia

<https://jansson.readthedocs.org/en/2.6/index.html>

<http://curl.haxx.se/libcurl/c/simple.html>

<http://curl.haxx.se/libcurl/c/sepheaders.html>

<http://curl.haxx.se/libcurl/c/getinmemory.html>

<http://www.cyberciti.biz/faq/delete-or-remove-a-directory-linux-command/>

<http://www.jsoneditoronline.org/>