

## 11.6 O Utilitário Unix. *Make* o Gestor do Projectos

O Utilitário *Make* é uma ferramenta do sistema operativo Unix.

É um gerador de comandos que é muito utilizado como uma ferramenta de engenharia de software para gestão e manutenção de programas.

*Make* funciona através um ficheiro de descrições que gere comandos para serem executados pelo Shell de Unix. É muito utilizado para gerir as *dependências* entre ficheiros.

- a criação dos ficheiros intermediários,
- a criação dum ficheiro executável,
- remoção de ficheiros temporários,
- instalação *linking*,
- criação de relatórios etc etc.

### Um Ficheiro Simples de Make

Por convenção o ficheiro de descrição tem nome *Makefile*  
O utilitário make chamado do shell de Unix leia esta ficheiro.

**> make program**

- ➔ criar uma versão, a mais nova, do "*program*".
- ➔ fazer todos os passos necessário de compilação e linking.

Exemplo : um programa para uma lista-ligada que tem:

- 3 ficheiros de código fonte c ➔ main.c iinterface.c listafn.c
- um ficheiro de assembler para rotinas de ordenação ➔ sort.s
- um conjunto de rotinas numa livraria ➔ /usr/projecto/biglib.a

Para criar o executável a sequência de comandos seria

> cc -c main.c

> cc -c iinterface.c

> cc -c listafn.c

> as -o sort.o sort.s

> cc -o lista main.o iinterface.o listafn.o sort.o /usr/projecto/biglib.a

```

# Makefile para listaligada
# O compilador de C , podia ser cc , gcc etc.
CC = cc
# O compilador de Assembly
AS = as

lista : main.o iointerface.o listafn.o sort.o o /usr/projecto/biglib.a
    $(CC) -o lista main.o iointerface.o listafn.o sort.o usr/projecto/biglib.a

main.o : main.c
    $(CC) -c main.c

iointerface.o : iointerface.c
    $(CC) -c iointerface.c

listafn.o : listafn.c
    $(CC) -c listafn.c

sort.o : sort.s
    $(AS) -o sort.o sort.s

# Limpar
limpar :
    /bin/rm -f core *.o

```

### Este ficheiro contem

- linhas de comentários ... texto que comece com o cardinal
- linhas de definição de variáveis var = valor.
- Linhas de Alvo (target) especificando as dependências, regras e comandos para executar.

O Alvo é o que nos pretendemos criar.

É separado das suas dependências através um colon. ∴

As linhas seguintes são os comandos. (começam com um tab )

*Portanto os comandos especifiquem como é que o alvo é criado a partir das suas dependências.*

## Por exemplo

- O alvo `main.o` é criado a partir do ficheiro `main.c` que é compilado com o comando `cc -c main.c`

Para criar este alvo > **make main.o**

- O alvo `limpar` não tem dependências, simplesmente execute o comando `rm -f` que vai remover os ficheiros `core` e `objecto`.

Para criar este alvo > **make limpar**

- O alvo `lista` depende em cinco ficheiros que quando existem são juntados com o linker via o comando `cc -o lista main.o iointerface.o listafn.o sort.o /usr/projecto/biglib.a`

Para criar este alvo > **make lista**

## Verificação das Dependências

A utilidade de *make* reside neste parte. Considere o caso em cima quando desejamos criara o executável `lista` através `make lista` e um dos ficheiros `objecto` não existe ou que um dos seus ficheiros código fonte tenha sido alterado e portanto um dos ficheiros `objecto` refere uma versão velha !

Make a ser invocado primeiro verifique as dependências do alvo !

i) se um dos ficheiros dos dependências não existe procure no resto das suas linhas para uma indicação como é que este ficheiro necessário pode ser criado.

ii) Para qualquer dependência verifique para ver se alguma dos dependência é mais nova do que a versão velha *lista*. Isto é fácil como Unix guarde sempre o ultimo tempo de modificação dum ficheiro.

iii) repete um processo recursivo para cada dependência, verificando cada dependência quando puder.

Make execute um mínimo número de comandos.. ver os exemplo a seguir

## CASO 1: todos os ficheiros fonte alterados

```
ciunix> make lista
cc -c main.c
cc -c iointerface.c
cc -c listafn.c
as -o sort.o sort.c
cc -o lista main.o iointerface.o listafn.o sort.o /usr/projecto/biglib.a
```

### CASO 2: apenas o ficheiro main.c alterado

```
ciunix> make lista
cc -c main.c
cc -o lista main.o iointerface.o listafn.o
```

### CASO 3 : ficheiro main.c alterado mas criado o ficheiro objecto )

```
ciunix> make lista
cc -o lista main.o iointerface.o listafn.o
```

### CASO 4 : nenhum ficheiro alterado .. não vale a pena recompilar !

```
ciunix> make lista
'lista' is up to date
```