# Automação de construção
# - make -

Lauro Moura
lauromoura@gmail.com

# Tópicos

- Introdução
- Makefile
- Funcionamento
- Alvos
- Dependências
- Regras
- Variáveis
- Exemplo
- Referências

# Introdução

- **make** é um programa para automação de construção de programas.
- A principal função do make é verificar que peças de um programa precisam ser recompiladas e então executar os comandos para atualizar essas peças.
- Pode ser utilizado com praticamente qualquer linguagem.

# Makefile

- O **makefile** é um arquivo que serve de base de dados para o make.
- Composto de conjuntos de **alvos**, **dependências** e **regras**.
- Formato:

alvo: dependências
<tab>regras

# Makefile (exemplo)

```
foo: doo.o zee.o
    gcc -o foo doo.o zee.o

doo.o: doo.asm
    nasm -felf doo.asm

zee.o: zee.c
    gcc -c zee.c
```

# Funcionamento

- De maneira geral, o make analisa as dependências de um alvo, verificando se elas estão atualizadas (recursivamente), e caso alguma dependência tenha mudado, reconstrói o alvo utilizando as regras correspondentes.

# Alvos

- Em ambientes C + Asm, os alvos normalmente são os executáveis e os arquivos-objeto fruto das compilações.
- Clean: um alvo especial é o alvo “clean”, que normalmente é usado para “reiniciar” o ambiente, apagando todos os arquivos-objeto e executáveis.

# Dependências

- Dependências são os arquivos dos quais os alvos dependem. Normalmente são arquivos-objeto ou arquivos-fonte.
- A comparação com o alvo se dá pela data de modificação.

# Regras

- Regras são os comandos utilizados para atualizar um alvo.
- Normalmente são idênticos aos comandos digitados no console.

# Variáveis

- O make suporta o uso de variáveis nos makefiles. Elas podem ser definidas tanto na linha de comando como dentro do makefile.
- Definição
  - <Nome> = <valor>
- Uso
  - $(<NOME>)

# Exemplo: Makefile

```
CC=gcc

fatorial: fatorial.o main.o
	$(CC) $(CFLAGS) -o fatorial fatorial.o main.o

fatorial.o: fatorial.asm
	nasm -felf fatorial.asm

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

clean:
	rm *.o fatorial
```

# Compilando

- $ make clean
  - Remove todos os alvos
- $ make CFLAGS='-O3 -g'
  - Compila com otimização e informações para debug.

# Referências

- GNU Make - GNU Project
- Man pages: man make
- Info pages: info make
- Advanced Linux Programming, Code Sorcery LLC (cap. 1 – Getting Started)
- Assembly Language Step by Step, 2nd Edition, Wiley