

```
guiao-6-0.1/src/  
guiao-6-0.1/src/Makefile.in  
guiao-6-0.1/src/Makefile.am  
guiao-6-0.1/src/Foo.java  
guiao-6-0.1/install.sh  
guiao-6-0.1/configure  
guiao-6-0.1/AUTHORS  
ubuntu@ubuntu:~/Desktop/guiao-6-auto$ ./configure  
checking for a BSD-compatible install... /usr/bin/install -c  
checking whether build environment is sane... yes  
checking for a thread-safe mkdir -p... /bin/mkdir -p  
checking for gawk... no  
checking for mawk... mawk  
checking whether make sets $(MAKE)... yes  
checking for style of include used by make... GNU  
checking for gcj... no  
checking dependency style of gcj... none  
checking for java... yes  
checking for javac... yes  
checking for pdflatex... yes  
checking for biber... yes  
checking Verificando existência do package biblatex... /usr/share/texmf/tex/latex/biblatex/biblatex.sty  
checking Verificando se é possível compilar programas Java... "Sim"  
configure: creating ./config.status  
config.status: creating Makefile  
config.status: creating src/Makefile  
config.status: creating doc/Makefile  
config.status: executing depfiles commands
```

http://blogs.msdn.com/blogfiles/willy-peter_schaub

AUTOMAÇÃO E AUTOCONFIGURAÇÃO

Automação de Tarefas

GNU Make

Compilação de Aplicações

- O código fonte de uma aplicação é dividido em ficheiros.
 - ▣ Tipicamente: um “módulo” por ficheiro

- Objectivos:
 - ▣ Facilitar compreensão
 - ▣ Melhorar cooperação em equipa
 - ▣ Facilitar modularidade
 - Possibilidade de trocar módulos

Compilação de Aplicações

- 1 programa de P1
 - ▣ 1 ficheiro^(*)
- 1 programa de P2
 - ▣ 4 ficheiros^(*)
- 1 programa na dissertação
 - ▣ 400 ficheiros^(*)

- Guião 5 de LABI: ~20 ficheiros

- Uma aplicação comercial:
 - ▣ Google Chrome: >21000 ficheiros
 - ▣ Linux: > 30000 ficheiros

Compilação de Aplicações

- Compilar projeto com 1 ficheiro:
 - ▣ `javac ficheiro.java`
- Compilar projeto com 20 ficheiros
 - ▣ 20 x `javac ficheiro.java`?
 - ▣ De cada vez que existem alterações...
- Compilar projeto com 400 ficheiros:
 - ▣ ??

Compilação de Aplicações

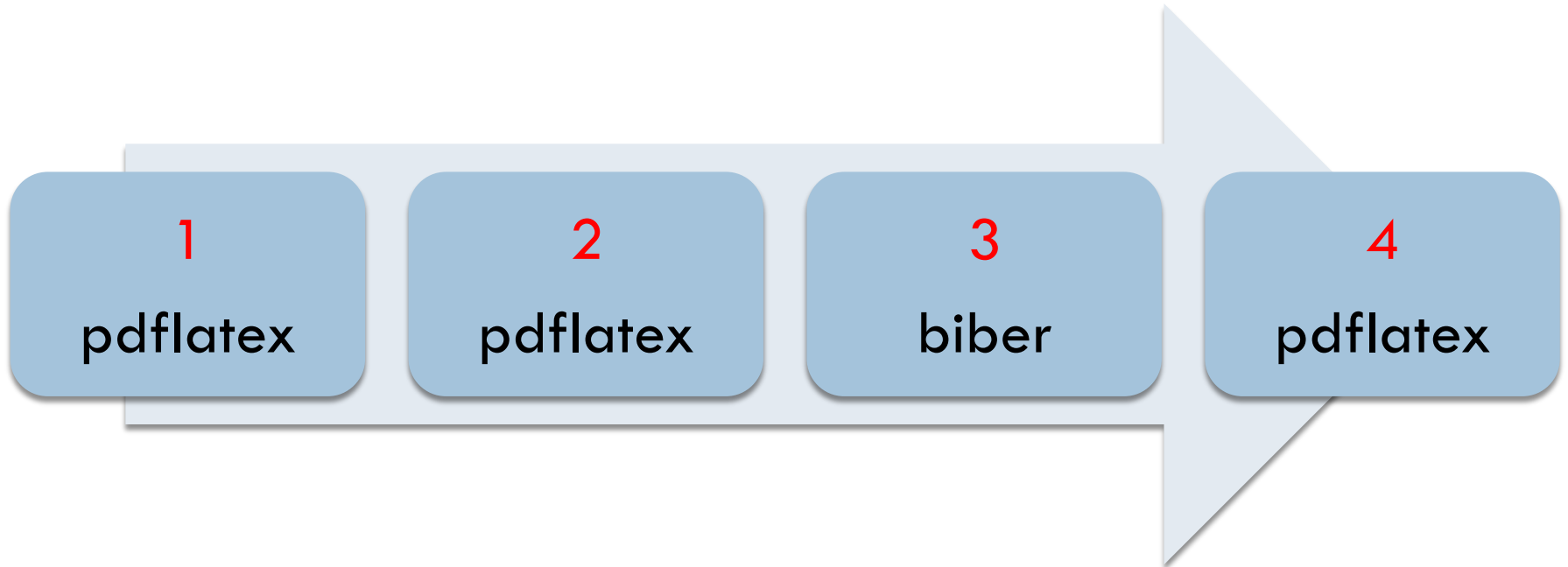
- Mesmo para um pequeno relatório em Latex
 - ▣ Com acrónimos, índices e bibliografia

- Passos necessário para compilação
 - ▣ Gerar acrónimos e TOC, LOF...: `pdflatex doc.tex`
 - ▣ Inserir acrónimos e TOC, LOF...: `pdflatex doc.tex`
 - ▣ Gerar citações: `biber doc`
 - ▣ Inserir citações: `pdflatex doc.tex`

- ▣ Opcional: limpar ficheiros temporários

Dependências

- Compilação raramente 100% paralelizável
- Por vezes até é necessariamente sequencial



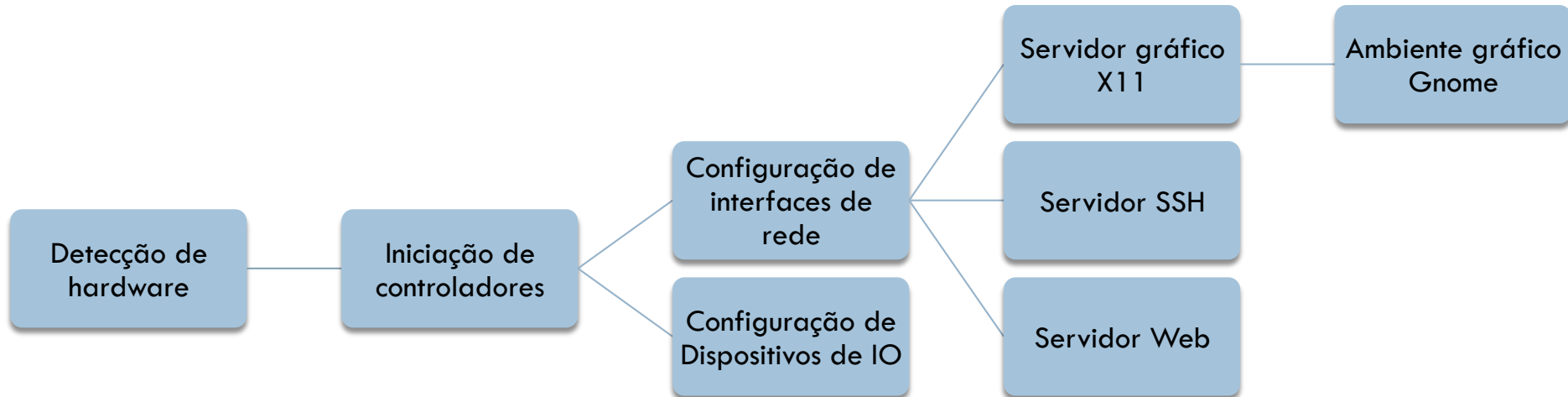
4 depende de 3, que depende de 2, que depende de 1

Automatização de Tarefas

- Permite facilitar tarefas repetitivas
 - ▣ Ex: comandos necessários para compilar Linux
 - make modules
 - make bzImage
- Foca-se na definição das ações...
 - ▣ Compilar projeto Foo
- ... e das suas dependências
 - ▣ Para realizar A, executa-se A1, mas primeiro necessário B

Automatização de Tarefas

- Conceito aplica-se a outras tarefas
 - ▣ Como iniciar todos os serviços no boot de um sistema?



GNU Make

- Apoia-se em ficheiros de definição de alvos e regras
 - ▣ Nome: Makefile
 - ▣ Ficheiro de texto
 - ▣ Normalmente 1 por directório
- Invocam-se acções executando make.
 - ▣ make all
 - ▣ make install
 - ▣ make clean
 - ▣ ...

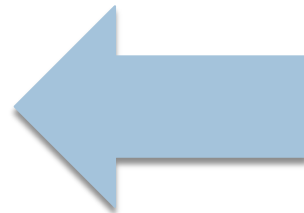
GNU Make

- Exemplo para 1 exercício da aula de P1

ex1.class:

javac ex1.java

all: ex1.class



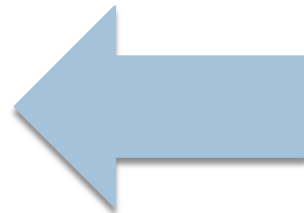
Conteúdo do ficheiro
Makefile

- Significado:
 - ▣ Alvo “all”, necessita de ficheiro ex1.class, que é construído executando javac ex1.java
- Executa-se: make all

GNU Make

- Exemplo para 2 exercícios da aula de P1

```
-----  
ex2.class:  
    javac ex2.java  
  
ex1.class:  
    javac ex1.java  
  
all: ex1.class ex2.class  
-----
```



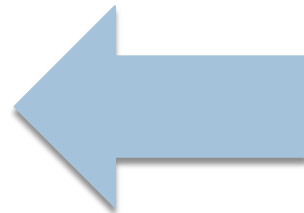
Conteúdo do ficheiro
Makefile

- Significado:
 - ▣ Alvo “all”, necessita de ficheiros ex1.class e ex2.class, que são construídos executando javac ex1.java e javac ex2.java
- Executa-se: make all

GNU Make

- Exemplo para todos os exercícios da aula de P1

```
-----  
%.class:  
    javac $*.java
```



Conteúdo do ficheiro
Makefile

```
all: ex1.class ex2.class ex3.class ex4.class  
-----
```

- Significado:
 - ▣ Alvo “all”, necessita de ficheiros ex1.class, ex2.class, ex3.class e ex4.class. Cada um é compilado com javac.
- Executa-se: make all

GNU Make

- Pode propagar ações para sub-directórios

```
AULAS = aula1 aula2 aula3
```

```
.PHONY $(AULAS)
```

```
$(AULAS) :
```

```
$(MAKE) -C $@
```

```
all: $(AULAS)
```

- Compilar aula1 aula2 e aula3: `make all`
 - ▣ Irá invocar `make all` em todos os sub-directórios

Autoconfiguração

GNU Autoconf

GNU Automake

Auto Configuração

Caso prático: Relatórios em LaTeX

- Os professores...
 - ▣ têm os packages/fontes que vocês utilizaram?
 - ▣ usam a mesma versão de latex?
 - Existe Tex, Latex2, Latex2e, Latex3, LuaTex, Xetex, ConTex...
 - ▣ sabem identificar qual o ficheiro principal?
 - ▣ sabem quantos passos são necessários para compilar?
 - ▣ Sabem qual o programa a utilizar para compilar?
 - biber vs bibtex
 - pdflatex, latex, etex, luatex, lualatex, pdftex, etc...

Auto Configuração

- Como distribuir uma aplicação em código fonte pela internet?
 - ▣ Problema semelhante ao do vosso relatório
 - ▣ Mas provavelmente com mais ficheiros
 - ▣ Mais dependências entre ficheiros e bibliotecas externas
- Solução: GNU Autoconf e Automake

GNU Autoconf

- Utiliza ficheiro “configure.ac”
 - ▣ Define regras de detecção de requisitos
 - ▣ Podem-se prever variações na compilação
 - Adaptação para diferentes SO
 - ▣ Pode efetuar testes para garantir compilação
- Ficheiro “configure.ac” convertido para um script “configure”
 - ▣ Efetua as detecções pretendidas
 - ▣ Deteta falha de requisitos
 - ▣ Configura sistema de compilação

GNU Autoconf: configure.ac

```
AC_INIT([src/Foo.java])
```

```
AM_INIT_AUTOMAKE([guião-6],[0.1])
```

```
AM_PROG_GCJ
```

```
AC_CHECK_PROG(EXISTE_JAVA,java,yes)
```

```
AC_CHECK_PROG(EXISTE_JAVAC,javac,yes)
```

```
AC_CHECK_PROG(EXISTE_PDFLATEX,pdflatex,yes)
```

```
AC_CHECK_PROG(EXISTE_BIBER,biber,yes)
```

GNU Autoconf: configure

```
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking for style of include used by make... GNU
checking for gcj... no
checking dependency style of gcj... none
checking for java... yes
checking for javac... yes
checking for pdflatex... yes
checking for biber... yes
```

configure.ac

...

```
AC_CHECK_PROG(EXISTE_JAVAC, javac, yes)
```

```
if test "$EXISTE_JAVAC"; then
```

```
    AC_MSG_NOTICE([Compilador de Java encontrado.])
```

```
else
```

```
    AC_MSG_ERROR([Compilador de Java em falta.])
```

```
fi
```

Resultado

```
checking for a BSD-compatible install... /usr/bin/install -c
```

```
checking whether build environment is sane... yes
```

```
checking for a thread-safe mkdir -p... /bin/mkdir -p
```

```
checking for gawk... no
```

```
checking for mawk... mawk
```

```
checking whether make sets $(MAKE)... yes
```

```
checking for javac... no
```

```
configure: error: Compilador de Java em falta.
```

GNU Automake

- Usa ficheiros Makefile.am
 - ▣ São convertidos para ficheiros Makefile
- Sintaxe semelhante a um ficheiro Makefile comum
 - ▣ Com adição de mais palavras chave
- Permite ao GNU Autoconf gerar ficheiros Makefile apropriados

Para Referência



- GNU Make: <http://www.gnu.org/software/make/>
- GNU Automake: <http://www.gnu.org/software/automake/>
- Gnu Autoconf: <http://www.gnu.org/software/autoconf/>