Dienstag: Makefiles

6 Makefiles

Building

- Beim *Build-Prozess* (Erstellungsprozess) wird Text in Binärdateien konvertiert.
- Meistens: Quellcode in selbständiges Programm kompilieren.
- Aber auch: PDF aus LaTeX-Dokument erstellen.
- Und: Archiv aus einem Projekt erstellen
- Im Allgemeinen: Eine Menge von Anweisungen, die oft in einer bestimmten Reihenfolge ausgeführt werden (z.B. `pdflatex, bibtex, pdflatex, pdflatex`)

Abhängigkeiten

- Erstellungsschritte müssen in einer bestimmten Reihenfolge ausgeführt werden.
- Haben sich nur Teile des Projekts geändert, müssen nur diese neu erstellt werden.
  1. file1.c → file1.o
  2. file2.c → file2.o
  3. file1.o, file2.o → program

Abhängigkeiten

- Erstellungsschritte müssen in einer bestimmten Reihenfolge ausgeführt werden.
- Haben sich nur Teile des Projekts geändert, müssen nur diese neu erstellt werden.
  1. file1.c $\rightarrow$ file1.o
  2. file2.c $\rightarrow$ file2.o
  3. file1.o, file2.o $\rightarrow$ program

Abhängigkeiten

- Erstellungsschritte müssen in einer bestimmten Reihenfolge ausgeführt werden.
- Haben sich nur Teile des Projekts geändert, müssen nur diese neu erstellt werden.
  1. file1.c $\rightarrow$ file1.o
  2. file2.c $\rightarrow$ file2.o
  3. file1.o, file2.o $\rightarrow$ program

Abhängigkeiten

- Erstellungsschritte müssen in einer bestimmten Reihenfolge ausgeführt werden.
- Haben sich nur Teile des Projekts geändert, müssen nur diese neu erstellt werden.
  1. file1.c $\rightarrow$ file1.o
  2. file2.c $\rightarrow$ file2.o
  3. file1.o, file2.o $\rightarrow$ program

Abhängigkeiten

- Erstellungsschritte müssen in einer bestimmten Reihenfolge ausgeführt werden.
- Haben sich nur Teile des Projekts geändert, müssen nur diese neu erstellt werden.
  1. file1.c $\rightarrow$ file1.o
  2. file2.c $\rightarrow$ file2.o
  3. file1.o, file2.o $\rightarrow$ program

Makefiles

6 Makefiles
- Make
- Apache Ant

Make

- Build-Management Tool
- Ermöglicht Definition von Abhängigkeiten
- Abhängigkeiten und Anweisungen werden in `Makefiles` geschrieben.
- Dokumentation:
  `http://www.gnu.org/software/make/manual/`

```
:~$ make <target>
```

Makefile

Struktur

```
target: prerequisities
	command
	command
```

Jede `command`-Zeile muss mit einem *tab* eingerückt sein!

Bestandteile

target (Ziel) oft Name der erzeugten Datei (z.B. `slides.pdf`)

prerequisites (Bedingungen) Dateien (oder andere Ziele), die im Voraus benötigt werden (z.B. `slides.tex`)

command (Anweisung) Wie ausgeführt, um Ziel zu erstellen (z.B. `pdflatex`)

Makefile-Beispiel

```
all: Token.class Tagger.class

# This is a comment
Token.class: Token.java
        javac Token.java

Tagger.class: Tagger.java
        javac Tagger.java

javadoc: Tagger.java Token.java
        javadoc *.java
```

Variablen

- Bedingungen können Variablen enthalten
- Diese werden am Anfang deklariert:

```
JAVAC=javac
```

- Und wie folgt verwendet:

```
Token.class: Token.java
        ${JAVAC} Token.java
```

Pattern-Regeln

- Bei größeren Projekten enthalten Makefiles oft viele Redundanzen.
- Pattern-Regeln definieren eine Regel für alle Dateien vom selben Typ.
- Pattern-Regel für Kompilierung von java Dateien:

```
%.class : %.java
	javac $<
```

(Einige) automatische Variablen

`$@`	Ziel der Regel
`$<`	Die erste Bedingung
`$?`	Alle Bedingungen, die neuer als das Ziel sind
`$^`	Alle Bedingungen

Mehr: `http://www.gnu.org/software/make/manual/html_node/Automatic-Variables.html`

Makefiles

6 Makefiles
- Make
- Apache Ant

Apache Ant

- Neueres *Build*-Werkzeug
- Konzentriert auf Java
- **`Makefile`** $\rightarrow$ **`build.xml`**
- XML-basiert
- Dokumentation: `http://ant.apache.org/manual/`

ant ausführen

- Kommandozeile: `:~$ ant [-buildfile file.xml]`
- Wenn `-buildfile` nicht angegeben ist, sucht `ant` im momentanen Verzeichnis nach einer Datei `build.xml`
- Option `-find [file]`: ant durchsucht alle übergeordneten Verzeichnisse nach `build.xml`
- Target angeben: `:~$ ant -buildfile file.xml target`
- Weitere Optionen `:~$ ant -help`

Vorteile von `ant`

- Plattformunabhängig
- Keine Tabs vor Befehlen $\rightarrow$ Weniger fehleranfällig.

Übung 7