

ftw.
Forschungszentrum Telekommunikation Wien
[Telecommunications Research Center Vienna]

CAMPARI-Trainings-Workshop Tools

Martina Umlauf
umlauft@ftw.at

Kplus
Erasmus-Programme

Zielgruppe und Vorkenntnisse

- Praktikanten CAMPARI und Interessierte
- Computer-Kenntnisse (Commandline)

© ftw. 2005 2

Kplus
Erasmus-Programme

Overview

- Zielgruppe & Vorkenntnisse
- vi Editor
- Make Utility

© ftw. 2005 3

Kplus
Erasmus-Programme

vi Editor - Philosophie

3 Modi:

- Command Mode
- Insert Mode
- Ex Mode

Starten:

- vi
- vi filename

© ftw. 2005 4

Kplus
Erasmus-Programme

Text einfügen

- a / A ... append / to end of line
- i / I ... insert / at beginning of line
- o / O ... open new line below / above

Zurück zum Command Mode:

- ESC

© ftw. 2005 5

Kplus
Erasmus-Programme

Text editieren

- x ... delete char under cursor
- cc ... change line
- cw ... change word
- c\$... change to end of line
- dd ... delete line
- dw ... delete word
- d\$... delete to end of line
- yy ... yank line ("copy")
- yw ... yank word
- y\$... yank to end of line
- p / P ... paste below / above

© ftw. 2005 6

Kplus
Erasmus-Programme

Cursor Movements



- \$... end of line
- % ... matching parantheses
- G ... goto
 - 5G ... goto line 5
 - 1G ... goto line 1
 - 0G ... goto last line

Suchen und Ersetzen



- /Suchbegriff ... Vorwärts suchen
- // ... Weitersuchen
- ?Suchbegriff ... Rückwärts suchen
- ?? ... Rückwärts weitersuchen
- n ... Letzte Suche in gleiche Richtung wiederholen
- N ... Letzte Suche in entgegengesetzte Richtung wiederholen
- :s/suchbegriff/ersatzbegriff/g
 - :1,\$s/falsch/richtig/g

Laden und Speichern



- vi filename
- vi file1 file2 file3 ...
- ZZ ... Speichern und beenden
- :q ... Quit
- :q! ... Quit ohne Speichern
- :wq ... Speichern, dann quit
- :w ... Zwischenspeichern
- :w filename ... Write to file filename
- :r filename ... Read filename

Multi-File Handling



- vi file1 file2 file3 ...
- :n ... Next file
- :N ... Prev file
- :rew ... Back to first file (rewind)

Diverse Befehle



- ^G ... Show current filename & status
- J ... Join lines (umgebrochene Zeilen verbinden)
- u ... Undo
- Repeat last command
- ! ... Execute shell command
 - !)fmt ... Absatz mit Unix fmt formatieren
- vi -r filename ... Abgebrochene Sitzung fortsetzen

Overview



- Zielgruppe & Vorkenntnisse
- vi Editor
- Make Utility

Make Utility



- Wiederkehrende Tätigkeiten mit File-Abhängigkeiten, zb.
 - Kompilieren
 - LaTeX
 - ...
- Makefile / makefile
- make
- make -f *file*
- make -n ... Nur display, nicht ausgeführt

© ftw. 2005

13



Makefile - Dependency Lines / 1



- Dependent Files -> Target
- Befehle zur Erzeugung des Targets

```
target: dependent files
command

```

Tab, NICHT Space

Leerzeile

```
target: dependent files
command1
command2
command3

target: dep. Files
command1; command2
```

© ftw. 2005

14



Makefile - Dependency Lines / 2



```
target: dependent files
command

makefile
myprog: main.o io.o
gcc main.o io.o -o myprog

main.o: main.c types.h
gcc -c main.c

io.o: io.c
gcc -c io.c

$ make
```

© ftw. 2005

15



Makefile - Dependency Lines / 3



```
makefile
print:
a2ps main.c io.c

install:
cp <...>

help:
@echo "print ... Print Sourcefiles"
@echo "install ... Install Files - needs root"

$make print
$make help
```

© ftw. 2005

16



Filestamps updaten



- Make wertet Filestamps aus
 - Löst Abhängigkeiten auf
 - Manchmal "Target is up to date"
- make "erzwingen":
- ```
$ touch main.c
```

© ftw. 2005

17



## Make Variablen



```
makefile
CFILES = main.c \
io.c
OBJ = main.o io.o
CFLAGS = -g -o myprog

myprog: $(OBJ)
gcc $(CFLAGS) $(OBJ)
```

© ftw. 2005

18



## Predefined Variablen



```
CC gcc
CFLAGS -O
LD ld
LDFLAGS
```

© ftw. 2005

19



## Special Variablen / 1



```
$$... Name des Target Files
$$? ... Dependent files more recent
 than the target
$$< ... Wie $$? in Suffix Rules
$$* ... Target ohne Suffix in Suffix Rules
```

© ftw. 2005

20



## Special Variablen / 2



### Makefile

```
print: main.c io.c
a2ps $$?
@touch print
```

← main.c, io.c wenn  
neuer als target

```
myprog: main.o io.o
gcc main.o io.o -o $$@
```

← myprog

© ftw. 2005

21



## Suffix Rules / 1



### Makefile

```
.suf1.suf2:
command
```

← Keine expliziten  
dependent files

```
.c.o:
```

```
gcc -c $$<
```

← zb. main.c

```
$$< ... Current dependent file inkl. Suffix
$$* ... Current target file inkl. Suffix
$$@ ... Current target file ohne Suffix
```

© ftw. 2005

22



## Suffix Rules / 2



### Makefile

```
.SUFFIXES: .bak
```

```
.c.bak:
```

```
cp $$< $$@
```

← zb. main.c

← zb. main.bak

© ftw. 2005

23



## Typische Targets / 1



```
$$ make all
```

### Makefile

```
all: myprog mylibs mydocs
```

```
myprog: main.o io.o
```

```
gcc main.o io.o -o $$@
```

© ftw. 2005

24



## Typische Targets / 2



```
$ make clean
```

### Makefile

```
clean:
 rm -f *.o myprog
```

## References



- **Mastering the vi Editor**, Ben Yoshino, University of Hawaii at Manoa, College of Engineering
- **VIVA vi!**, Carl de Boor, wisc.edu
- **vi tutor**, Michael C. Pierce, Robert K. Ware, Carl de Boor
- **Managing Projects with make**, Andrew Oram, Steve Talbott, O'Reilly, ISBN 0-937175-90-0
- **Linux Programmer's Reference**, Richard Petersen, Osborne, ISBN 0-07-882587-3

## Folgetermine



- **Di, 23. Aug, 9:00-12:00**  
Extreme Programming light and good development practices
- **Fr, 26. Aug, 9:00-12:00**  
offen, wahrscheinlich Kurzvortrag Python

## The End



**Danke für die  
Aufmerksamkeit!**