Computersystemen 2
Labo
Compileren

Inhoud

- Compileren
- Make

Inhoud

- Compileren

Compileren

- gcc -o programma programma.c
- g++ -o programma programma.cpp
- … zie theorie gedeelte.

Inhoud

- Make

Source files

- 4 files:
  - Main.cpp = source file
  - Helloworld.cpp
  - Factorial.cpp
  - Functions.h = header file

Source files

- 4 files:
  - Main.cpp = source file
  - Helloworld.cpp
  - Factorial.cpp
  - Functions.h = header file

```
#include <iostream>
#include "functions.h"
int main(){
print_hello();
std::cout << std::endl;
std::cout << "De faculteit van
10 is " << factorial(10) <<
std::endl;
return 0;
}
```

Source files

- 4 files:
  - Main.cpp = source file
  - Helloworld.cpp
  - Factorial.cpp
  - Functions.h = header file

```
#include <iostream>
#include "functions.h"
void print_hello(){
std::cout << "Hello World!";
}
```

Source files

- 4 files:
  - Main.cpp = source file
  - Helloworld.cpp
  - Factorial.cpp
  - Functions.h = header file

```
#include "functions.h"
int factorial(int n){
if(n!=1){
return(n * factorial(n-1));
}
else return 1;
}
```

Source files

- 4 files:
  - Main.cpp = source file
  - Helloworld.cpp
  - Factorial.cpp
  - Functions.h = header file

```
void print_hello();
int factorial(int n);
```

Manueel compileren…

- g++ main.cpp helloworld.cpp factorial.cpp -o helloworld

Compile met make…

- Sudo apt-get install make
- make zoekt naar “Makefile” in source bestand

Compile met make...

- Makefile: bevat compileer-instructies.
- Basis:

target: dependencies
[tab] systeemcommando's

```
all:
g++ main.cpp helloworld.cpp
factorial.cpp -o helloworld
```

Compile met make...

- Makefile Met dependencies...

```
all: helloworld

helloworld: main.o factorial.o helloworld.o
	g++ main.o factorial.o helloworld.o -o helloworld

main.o: src/main.cpp
	g++ -c src/main.cpp

factorial.o: src/factorial.cpp
	g++ -c src/factorial.cpp

helloworld.o: src/helloworld.cpp
	g++ -c src/helloworld.cpp

clean:
rm -rf *o helloworld
```

Compile met make...

- Makefile Met variabelen en

```
# Dit is commentaar
# De variabele CC bevat de compiler die gebruikt zal worden
CC=g++
# De variabele CFLAGS bevat de compileopties
CFLAGS=-c -Wall

all: helloworld

helloworld: main.o factorial.o helloworld.o
    $(CC) main.o factorial.o helloworld.o -o helloworld
```

```
main.o: main.cpp
    $(CC) $(CFLAGS) main.cpp

factorial.o: factorial.cpp
    $(CC) $(CFLAGS) factorial.cpp

helloworld.o: helloworld.cpp
    $(CC) $(CFLAGS) helloworld.cpp

clean:
    rm -rf *o helloworld
```

Wat is er anders tov vorige

Compile met make...

- Makefile Met variabelen en

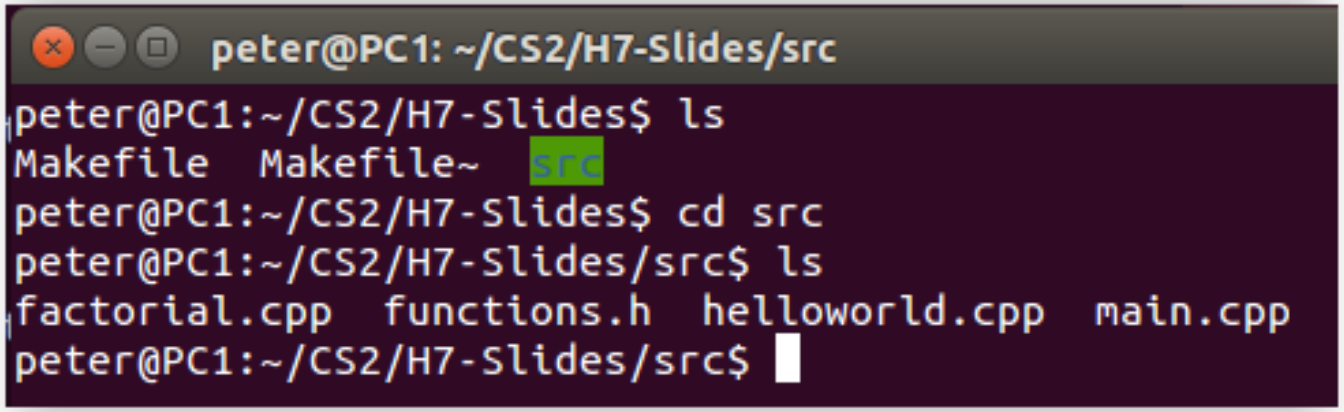
```
# Dit is commentaar
# De variabele CC bevat de compiler die gebruikt zal worden
CC=g++
# De variabele CFLAGS bevat de compileopties
CFLAGS=-c -Wall

all: helloworld

helloworld: main.o factorial.o helloworld.o
    $(CC) main.o factorial.o helloworld.o -o helloworld
```

```
main.o: main.cpp
    $(CC) $(CFLAGS) main.cpp

factorial.o: factorial.cpp
    $(CC) $(CFLAGS) factorial.cpp

helloworld.o: helloworld.cpp
    $(CC) $(CFLAGS) helloworld.cpp

clean:
    rm -rf *o helloworld
```

Wat is er fout tov vorige slide?
De /src zijn verdwenen…
(Gaat er dus van uit dat de sources in de huidige dir

Compile met make...

- Makefile Met variabelen en commentaar
- We testen of het werkt:



Compile met make…

- Makefile Met variabelen en commentaar
- We testen of het werkt:

```
g++ -c -Wall src/main.cpp
g++ -c -Wall src/factorial.cpp
g++ -c -Wall src/helloworld.cpp
g++ main.o factorial.o helloworld.o -o helloworld
peter@PC1:~/CS2/H7-Slides$
```

Compile met make…

- Makefile Met variabelen en commentaar
- We testen of het werkt:
  - Commando = make
  - Optioneel Commando = make helloworld

```
peter@PC1:~/CS2/H7-Slides$ ls
factorial.o  helloworld  helloworld.o  main.o  Makefile  Makefile~  src
peter@PC1:~/CS2/H7-Slides$
```

Compile met make…

- Makefile Met variabelen en commentaar
- We testen of het werkt:

```
peter@PC1:~/CS2/H7-Slides$ ls
factorial.o  helloworld  helloworld.o  main.o  Makefile  Makefile~  src
peter@PC1:~/CS2/H7-Slides$
```

```
peter@PC1:~/CS2/H7-Slides$ ./helloworld
Hello World!
De faculteit van 10 is 3628800
peter@PC1:~/CS2/H7-Slides$
```

Manpage

- Debian Policy: manpage nodig voor elke executable.
- Genereren met gmanedit
- Install: sudo apt-get install gmanedit
- Gebruik: sudo gmanedit
  - Anders gewone texteditor

  Zie: http://www.ghacks.net/2010/06/04/edit-linux-man-pages-with-gmanedit/

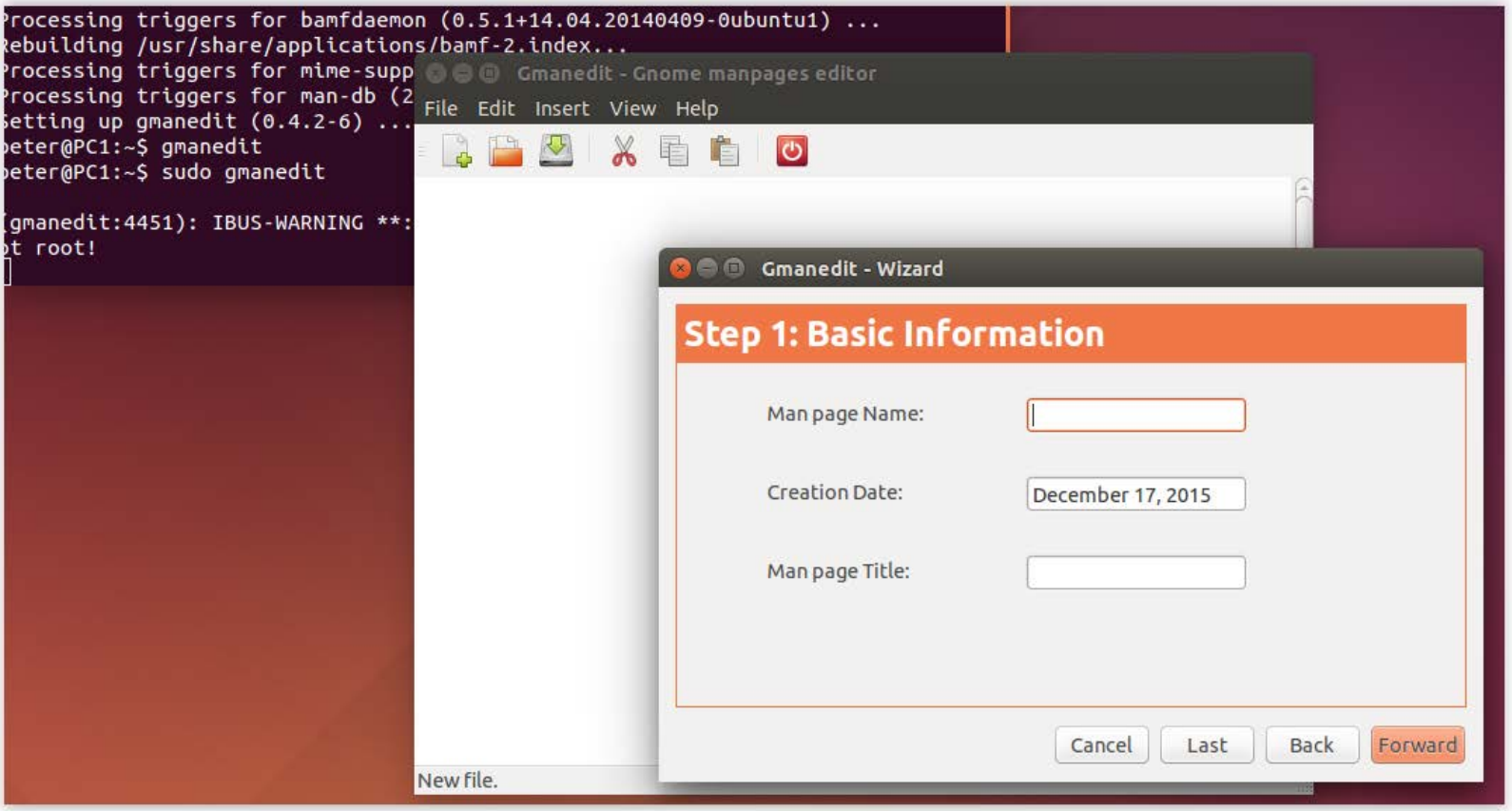
Manpage

- Manpage sections:

Nummer	Sectie	Uitleg
1	User command	Uitvoerbare commando's en scripts
2	System calls	Kernel functies
3	Library calls	Functies uit systeembibliotheken
4	Special files	Meestal bestanden in /dev
5	File formats	Bijvoorbeeld formaat /etc/passwd
6	Games	Spelletjes of andere frivoliteiten
7	Macro packages	Bijvoorbeeld man macro's
8	System administration	Programma's die enkel door root worden uitgevoerd
9	Kernel routines	Niet-standaard aanroepen en interne kernel functies

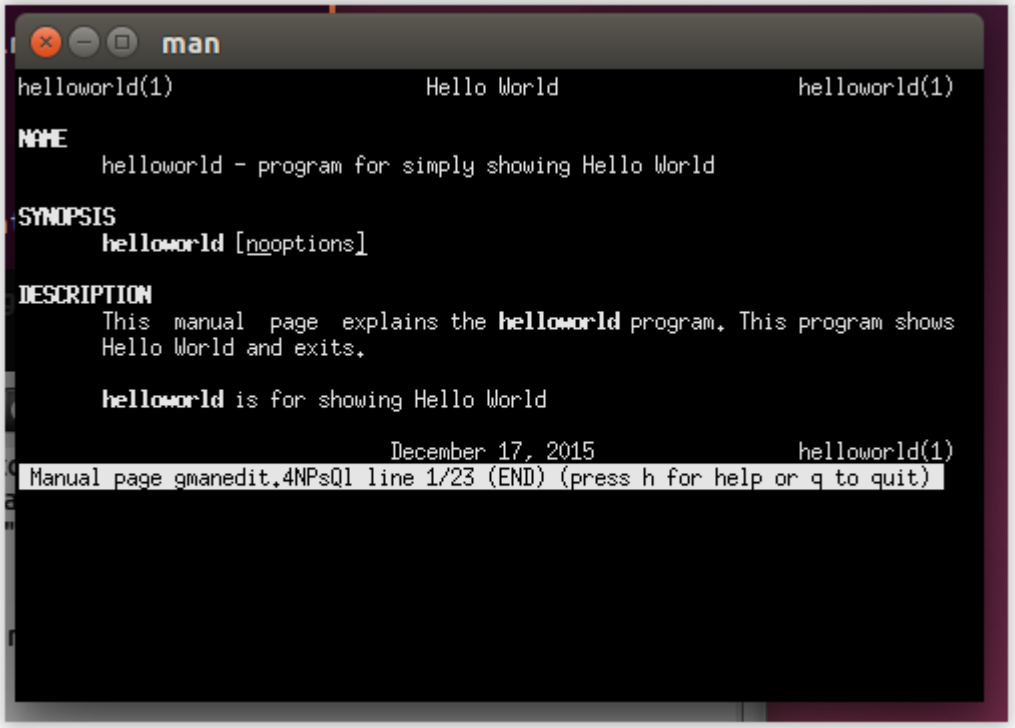
Manpage

- Manpage maken:
  - Wizard



Manpage

- Manpage maken: “View Created Page”



Manpage

- Manpage maken: (vb)

```
.TH helloworld 1 "February 25, 2014" "" "helloworld"

.SH NAME
helloworld \- program for displaying Hello World on the screen

.SH SYNOPSIS
.B helloworld
.RI [ no options ]
.br

.SH DESCRIPTION
This manual page explains how helloworld works. The program displays a
friendly Hello World message on the screen and exits.
.B helloworld
program. This program shows Hello World
.PP
\fBhelloworld\fP is for showing Hello World

.SH EXAMPLE
helloworld
Hello World
```

Manpage

- Locatie:
  - The system stores its man pages at /usr/share/man/ directory as described in about section. For example, the directory /usr/share/man/man1 stores man pages for user shell commands.
  - It is recommended that you store your own man pages in

    /usr/local/man directory.

```
/usr/local/share/man/man1/helloworld.1
peter@PC1:~/CS2$
```

The end…