10. Extrafolien zu „make“

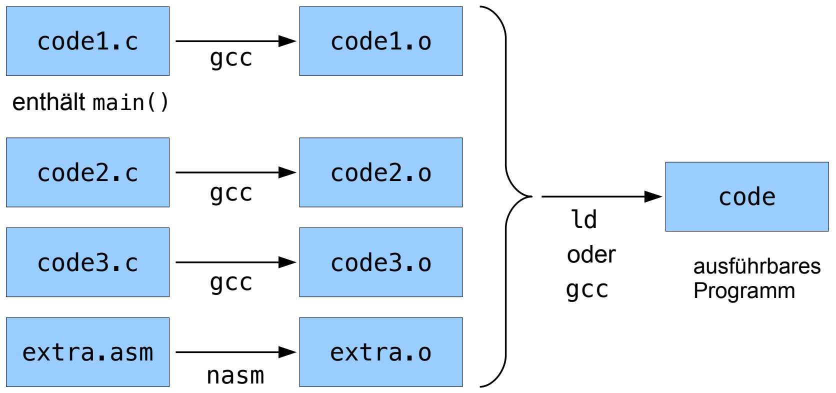
```
Sep 19 14:20:18 amd64 sshd[20494]: Accepted rsa for esser from ::ffff:87.234.201.207 port 61557
Sep 19 14:27:41 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 01:00:01 amd64 /usr/sbin/cron[29278]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 20 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 02:00:01 amd64 /usr/sbin/cron[30103]: (root) CMD (/sbin/evlogmgr -c 'age > "30d"')
Sep 20 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 12:46:44 amd64 sshd[6516]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62004
Sep 20 12:46:44 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 12:48:41 amd64 sshd[6609]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62105
Sep 20 12:54:44 amd64 sshd[6694]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62514
Sep 20 15:27:35 amd64 sshd[9077]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64242
Sep 20 15:27:35 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 16:37:11 amd64 sshd[10102]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63375
Sep 20 16:37:11 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 20 16:38:10 amd64 sshd[10140]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63546
Sep 21 01:00:01 amd64 /usr/sbin/cron[[illegible]055]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 21 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 02:00:01 amd64 /usr/sbin/cron[[illegible]878]: (root) CMD (/sbin/evlogmgr -c 'age > "30d"')
Sep 21 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 17:43:26 amd64 sshd[31088]: Accepted rsa for esser from ::ffff:87.234.201.207 port [illegible]397
Sep 21 17:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 17:53:39 amd64 sshd[31269]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64391
Sep 21 18:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 21 19:43:26 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 01:00:01 amd64 /usr/sbin/cron[4674]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 22 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 02:00:01 amd64 /usr/sbin/cron[5499]: (root) CMD (/sbin/evlogmgr -c 'age > "30d"')
Sep 22 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 22 20:23:21 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 01:00:01 amd64 /usr/sbin/cron[24739]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 23 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 02:00:01 amd64 /usr/sbin/cron[25555]: (root) CMD (/sbin/evlogmgr -c 'age > "30d"')
Sep 23 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 18:04:05 amd64 sshd[6554]: Accepted publickey for esser from ::ffff:192.168.1.5 port 59771 ssh2
Sep 23 18:04:05 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 23 18:04:34 amd64 sshd[6606]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62093
Sep 24 01:00:01 amd64 /usr/sbin/cron[12436]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 24 01:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 02:00:01 amd64 /usr/sbin/cron[13253]: (root) CMD (/sbin/evlogmgr -c 'age > "30d"')
Sep 24 02:00:01 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 11:15:48 amd64 sshd[20998]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64456
Sep 24 11:15:48 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 13:49:08 amd64 sshd[23197]: Accepted rsa for esser from ::ffff:87.234.201.207 port 61330
Sep 24 13:49:08 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 15:42:07 amd64 kernel: snd_seq_midi_event: unsupported module, tainting kernel.
Sep 24 15:42:07 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 24 15:42:07 amd64 kernel: snd_seq_oss: unsupported module, tainting kernel.
Sep 24 20:25:31 amd64 sshd[29399]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62566
Sep 24 20:25:31 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 01:00:02 amd64 /usr/sbin/cron[662]: (root) CMD (/sbin/evlogmgr -c "severity=DEBUG")
Sep 25 01:00:02 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 02:00:01 amd64 /usr/sbin/cron[1484]: (root) CMD (/sbin/evlogmgr -c 'age > "30d"')
Sep 25 02:00:02 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 10:59:25 amd64 sshd[8889]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64183
Sep 25 10:59:25 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 10:59:47 amd64 sshd[8921]: Accepted rsa for esser from ::ffff:87.234.201.207 port 64253
Sep 25 11:30:02 amd64 sshd[9372]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62029
Sep 25 11:59:25 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 14:05:37 amd64 sshd[11554]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62822
Sep 25 14:05:37 amd64 syslog-ng[7653]: STATS: dropped 0
Sep 25 14:06:10 amd64 sshd[11586]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62951
Sep 25 14:07:17 amd64 sshd[11608]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63392
Sep 25 14:08:33 amd64 sshd[11630]: Accepted rsa for esser from ::ffff:87.234.201.207 port 63709
Sep 25 15:25:33 amd64 sshd[12930]: Accepted rsa for esser from ::ffff:87.234.201.207 port 62778
```

GNU make

- Standard-Tool unter Linux/Unix für das automatische Übersetzen von Software
- setzt voraus, dass eine Regeldatei „Makefile“ vorhanden ist
- Regel-basiertes Kompilieren von Quellcode- und Binden von Objekt-Dateien
- Beispieldateien in ue09.tgz

GNU make

- Beispiel mit C- und Assembler-Dateien



Makefiles

- (einfaches) Makefile für das Beispiel:

```
CC=/usr/bin/gcc
LD=/usr/bin/ld
CFLAGS=-O3
ASM=nasm
ASMFLAGS=-felf32
LDFLAGS=

%.o: %.c
   $(CC) -c $(CFLAGS) $< -o $@

%.o: %.asm
   $(ASM) $(ASMFLAGS) $<

code: code1.o code2.o code3.o extra.o
   $(CC) $(LDFLAGS) -o code $^
```

Aufteilung des Quellcode

- Typisch: zu jeder .c-Datei eine .h-Datei, welche die Prototypen enthält.
- Im Beispiel:
  - `code1.c` : Implementation von `main()`
  - `code2.c` : Implementation von `get_a()`
  - `code2.h` : Protoyp für `get_a()`
  - `code3.c` : Implementation von `get_b()`
  - `code3.h` : Protoyp für `get_b()`
  - `extra.asm` : Assembler-Code

code1.c

```
#include <stdio.h>
#include "code2.h"
#include "code3.h"

extern int assembler();
              // aus extra.asm

int main () {
  int a, b, c;
  a = get_a ();
  b = get_b ();
  c = assembler ();
  printf ("a=%d, b=%d, c=%d\n",
          a, b, c);
}
```

code2.h

```
int get_a ();
```

code2.c

```
int get_a () {
  return 'A';   // 65
}
```

code3.h

```
int get_b ();
```

code3.c

```
int get_b () {
  return 'B';   // 66
}
```

extra.asm

```
global assembler

assembler: mov eax, 42
           ret
```

```
fom@fomdebian:~/sp/ue09$ make
/usr/bin/gcc -c -O3 code1.c -o code1.o
/usr/bin/gcc -c -O3 code2.c -o code2.o
/usr/bin/gcc -c -O3 code3.c -o code3.o
nasm -felf32 extra.asm
/usr/bin/gcc  -o code code1.o code2.o
  code3.o extra.o
fom@fomdebian:~/sp/ue09$ ./code
a=65, b=66, c=42
```

Regeln im Makefile

- Genereller Aufbau:

```
target: prerequisite(s)
   action
```

(Einrückung mit [Tab], nicht Leerzeichen!)

- Beispiel:

```
%.o: %.c
   $(CC) -c $(CFLAGS) $< -o $@
```

  - Target: `%.o` (z. B. `code2.o`)
  - Prequisite: `%.c` (z. B. `code2.c`)
  - `$<` = Quelldatei, `$@` = Zieldatei

Variablen im Makefile

- legen Compiler, Assembler, Linker und andere Tools sowie Optionen fest

```
CC=/usr/bin/gcc
LD=/usr/bin/ld
CFLAGS=-O3
ASM=nasm
ASMFLAGS=-felf32
LDFLAGS=
```

- Verwendung durch $(...)-Notation:

```
$(CC) -c $(CFLAGS) $< -o $@
$(ASM) $(ASMFLAGS) $<
$(CC) $(LDFLAGS) -o code $^
```