

C Build Tools

GCC

... that's it, isn't it?

No!!

Building C code

Tweak the compiler

Cool tools

Building C code

Preprocessor

Create objects files

Link

C preprocessor

Before compilation your source code is processed (before - pre, processed - processor). Basically it is a text substitution on lines beginning with #.

```
#define MAX_STUDENTS 100
```

```
#include <stdio.h>
```

Try:

```
gcc -E somefile.c
```

Create object files

From the preprocessed file the compiler creates object files (.o).

Try the following:

```
gcc -c somefile.c
```

Link

Put together one or many objects files into a program (binary).

```
gcc apa.o bepa.o cepa.o
```


Tweak the compiler

-c Only compile, do not link

-E Halt after preprocessing

-S “The output is in the form of an assembler code file for each non-assembler input file specified.” GCC Manual

-v Verbose mode

-o <file> Store output in file

-Wall This enables all the warnings about constructions that some users consider questionable,

-Werror Make all warnings into hard errors. Source code which triggers warnings will be rejected.

Inspecting C code

The `file` command helps us determine the type of a file.

```
hesa@schnittke:~/tmp/C$ file print.c
```

```
print.c: C source, ASCII text
```

```
hesa@schnittke:~/tmp/C$ file print.h
```

```
print.h: ASCII text
```

Inspecting C code

The `file` command helps us determine the type of a file.

```
hesa@schnittke:~/tmp/C$ file print.o
```

```
print.o: ELF 64-bit LSB relocatable, x86-64, version 1 (SYSV), not stripped
```

```
hesa@schnittke:~/tmp/C$ file a.out
```

```
a.out: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically  
linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.32,  
BuildID[sha1]=bf9871a0e656600bba3565ffe98863cd38e137fc, not stripped
```

Inspecting C code

The `objdump` command displays info from object files.

```
hesa@schnittke:~/tmp/C$ objdump -t print.o
```

```
print.o:    file format elf64-x86-64
```

```
SYMBOL TABLE:
```

```
0000000000000000 l   df *ABS*      0000000000000000 print.c
```

```
....
```

```
0000000000000000 g    F .text      000000000000002c verbose
```

Inspecting C code

The `objdump` command displays info from object files.

```
hesa@schnittke:~/tmp/C$ objdump -t a.out
```

```
a.out:          file format elf64-x86-64
```

```
SYMBOL TABLE:
```

```
0000000000400238 1      d  .interp      0000000000000000      .interp
0000000000400254 1      d  .note.ABI-tag  0000000000000000      .note.ABI-tag
0000000000400274 1      d  .note.gnu.build-id 0000000000000000      .note.gnu.build-
id
```

```
..... . .
```

Cool tools

`strace` - trace system calls and signals

`time` - run programs and summarize system resource usage

`gcov` - coverage testing tool

`gprof` - display call graph profile data

....

`gdb` - later on in the course

`valgrind` - later on in the course

Makefile

Makefile(s) are text files written in a certain prescribed syntax.

Together with **Make** Utility, it helps build a software from its source files, a way to organize code, and its compilation and linking.

<https://en.wikipedia.org/wiki/Makefile>