

Linking and Makefiles

- Chapter 7
 - Loading precompiled “trusted” code
 - Stuff you don’t have to understand

Looking back at LibMath

- `-lm /lib/libmath.so`
 - `gcc -m32 -o test_foo test.c /lib/libm.so`
 - `ldd test_foo`
 - `linux-gate.so.1 (0xf7ed5000)`
 - `libm.so.6 => /lib/libm.so.6 (0xf7ded000)`
 - `libc.so.6 => /lib/libc.so.6 (0xf7bfa000)`
 - `/lib/ld-linux.so.2 (0xf7ed7000)`

Symbols in Debug

- gdb test_foo
 - *gcc -g -m32 -o test_foo test.c /lib/libm.so*
- Without “-g” no symbols exist
 - Hints for the gnu debugger

Symbols in Debug

- Code listing

```
(gdb) list
1      #include <stdio.h>
2      #include <values.h>
3
4      int main()
5      {
```

Simple Makefiles

- CC → What compile toolchain
 - GCC mostly
- CFLAGS → Controls options
 - -g -m32 -Wall
 - -g → debug symbols
 - -m32 → machine type 32 bit
 - -Wall → all the warnings

Simple Makefiles

- `all: PROGRAM`
 - GNU Make target to handle building your program
- `PROGRAM:`
 - All the objects that need to be made
 - `LDFLAGS` and libraries

Simple Makefiles

- PROGRAM.o
 - Target to list all the source files to turn into objects
 - Uses CFLAGS to build
- .PHONY → mainly a clean target
 - Exists because clean doesn't have "Dependencies"

Simple Makefiles

```
CC=gcc

CFLAGS=-g -m32 -Wall

LDFLAGS=-m32 -lm

all: test_foo

test.o: test.c
    ${CC} ${CFLAGS} -c $< -o $@

test_foo: test.o
    ${CC} ${LDFLAGS} $< -o $@

.PHONY: clean

clean:
    rm *.o ${PROGRAM}
```

```
[sbruno@s1066246 ~]$ make
gcc -g -m32 -Wall -c test.c -o test.o
gcc -m32 -lm test.o -o test_foo
[sbruno@s1066246 ~]$ make clean
rm *.o
[sbruno@s1066246 ~]$ make clean
rm *.o
rm: cannot remove '*.o': No such file or directory
make: *** [Makefile:18: clean] Error 1
[sbruno@s1066246 ~]$
```

Simple Makefiles

- Dependencies
- Why is this needed?

```
[sbruno@s1066246 ~]$ make clean
rm -f *.o
[sbruno@s1066246 ~]$ make
gcc -g -m32 -Wall -c test.c -o test.o
gcc -m32 -lm test.o -o test_foo
[sbruno@s1066246 ~]$ make
make: Nothing to be done for 'all'.
[sbruno@s1066246 ~]$ touch test.c
[sbruno@s1066246 ~]$ make
gcc -g -m32 -Wall -c test.c -o test.o
gcc -m32 -lm test.o -o test_foo
[sbruno@s1066246 ~]$ █
```