Compiling, linking,...

- Let's see what happens when we compile a program using gcc
- Some of this material is based on section 7 of Computer Systems, A Programmer's Perspective, by Bryant O'Hallaron.

```
/* main.c */
void swap();
int buf[2] = {1,2};
int main()
{
      swap();
      return 0;
 }
```

```
/* swap.c */
extern int buf[]
int *bufp0 = &buf[0];
int *bufp1;
{
      int temp;
      bufp1 = &buf[1];
      temp = *bufp0;
      *bufp0 = *bufp1;
      *bufp1 = temp;
}
```

Compiling, linking,...

- What happens when you use the *compiler driver* gcc:

  gcc -O2 -g -o test main.c swap.c

Step1: Preprocessor (cpp)

- The C preprocessor translates the C source file *main.c* into an ASCII intermediate file *main.i*
- cpp [other args] main.c /tmp/main.i

Step2:
C compiler (cc1)

- Driver runs C compiler cc1
  - translates *main.i* into an ASCII assembly language file *main.s*
- cc1 /tmp/main.i main.c -O2 [other args] -o /tmp/main.s

Step3: assembler (as)

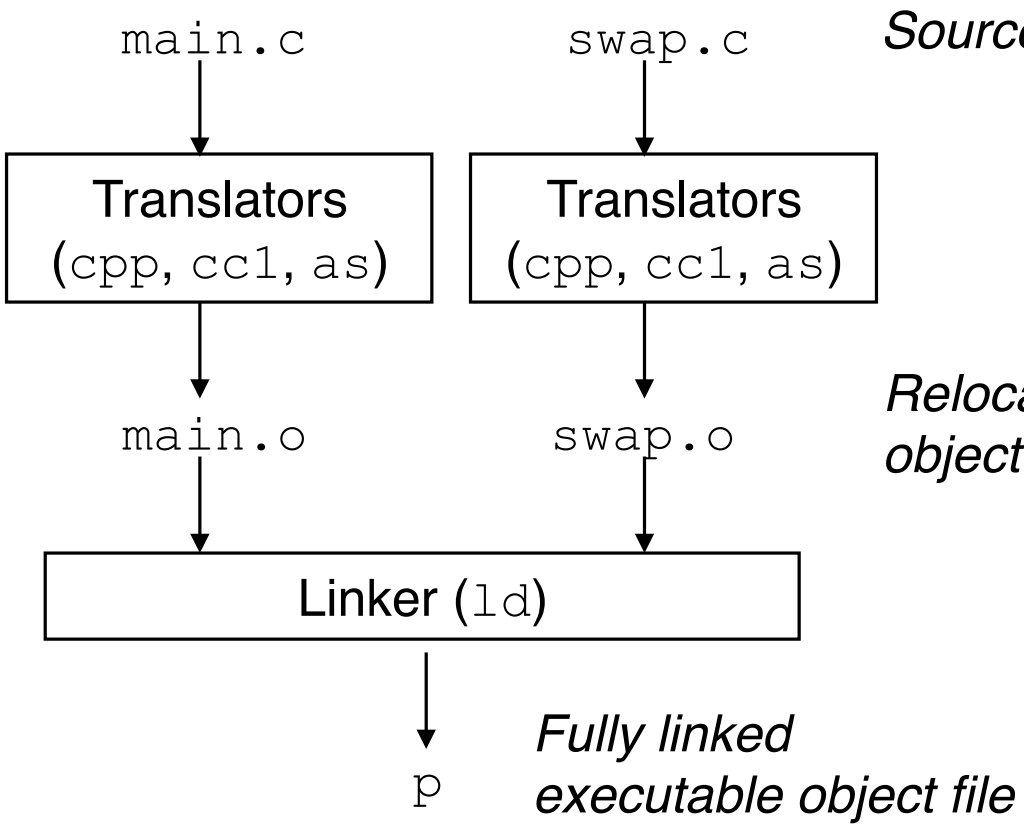
- driver runs assembler *as* to translate *main.s* into relocatable object file *main.o*
- as [other args] -o /tmp/main.o /tmp/main.s

- Same 3 steps are executed for swap.c

Step4:
Linker (ld)

- driver runs linker *ld* to combine *main.o* and *swap.o* into the executable file *test*
- ld -o test [sys obj files and args] /tmp/main.o / tmp/swap.o

Overview


main.c
swap.c
Source files
Translators
(cpp, cc1, as)
Translators
(cpp, cc1, as)
main.o
swap.o
Relocatable
object files
Linker (ld)
p
Fully linked
executable object file

Run executable *test*

- unix> ./test
- shell invokes function in OS called *loader*
  - copy code and data of executable *test* into memory
  - then transfers control to the beginning of the program

Static linking

- What does the static linker all do?
  - takes as input a collection of relocatable object files and command-line arguments
  - generates fully linked executable object file that can be loaded and run

Static linking

- Linker performs two tasks:
  - symbol resolution: associates each symbol reference with exactly one symbol definition
  - relocation: compiler and assembler generate code and data sections that start at address 0. Linker relocates these sections: associates memory location with each symbol definition, then updates all references to point to these locations.

3 Types of Object Files

- file.o
  - Relocatable object file
  - Executable object file
  - Shared object file (libraries)

Object Files

- Relocatable object file

  contains binary code and data in a form that can be combined with other relocatable object files at compile time to create an executable object file

Object Files

- Shared object file (libraries)

  special type of relocatable object file that can be loaded into memory and linked dynamically, either at load time or run time.

Object Files

- Executable object file

  contains binary code and data that can be copied directly into memory and executed

Object Files

- Putting it all together:
    - Compiler and assembler generate relocatable object files, including shared object files.
    - Linker generates executable object files.

Object Files

- Object formats vary
  - first Unix systems used a.out format (the term is still used)
  - early System V used Common Object File format (COFF)
  - Linux and other modern unix systems use Executable and Linkable Format (ELF).