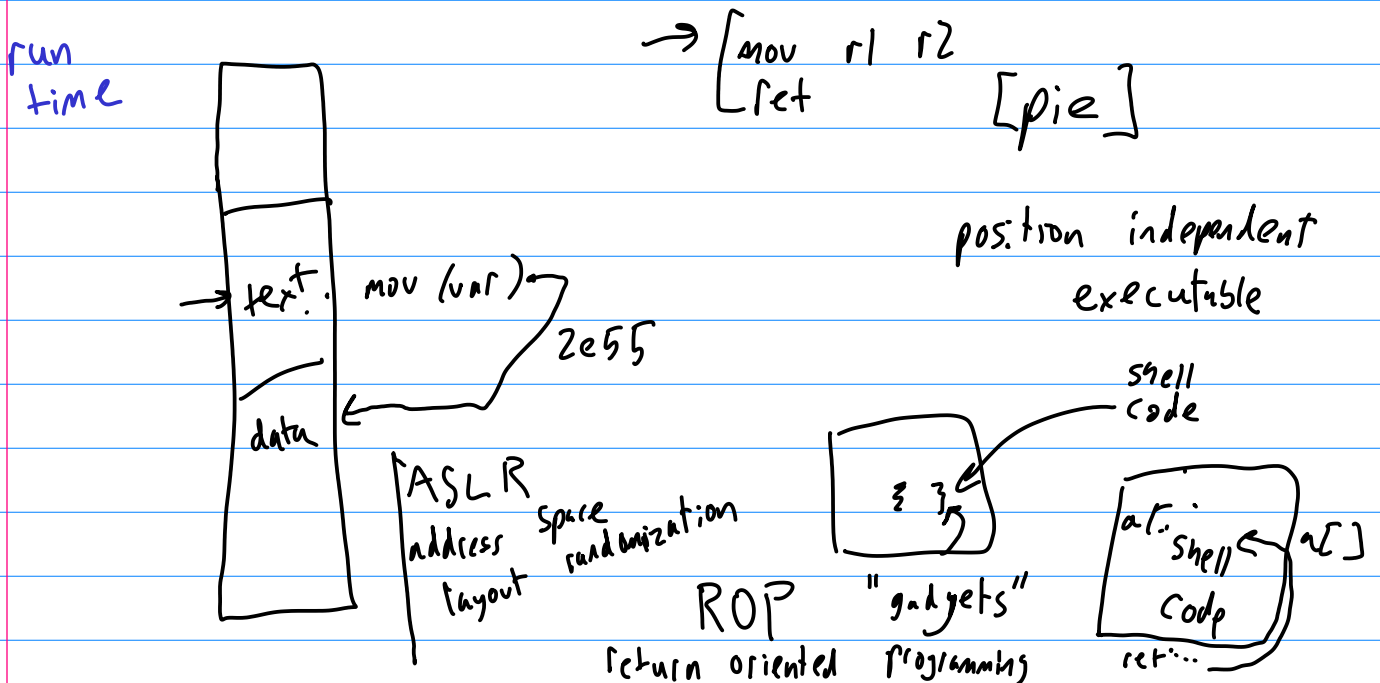
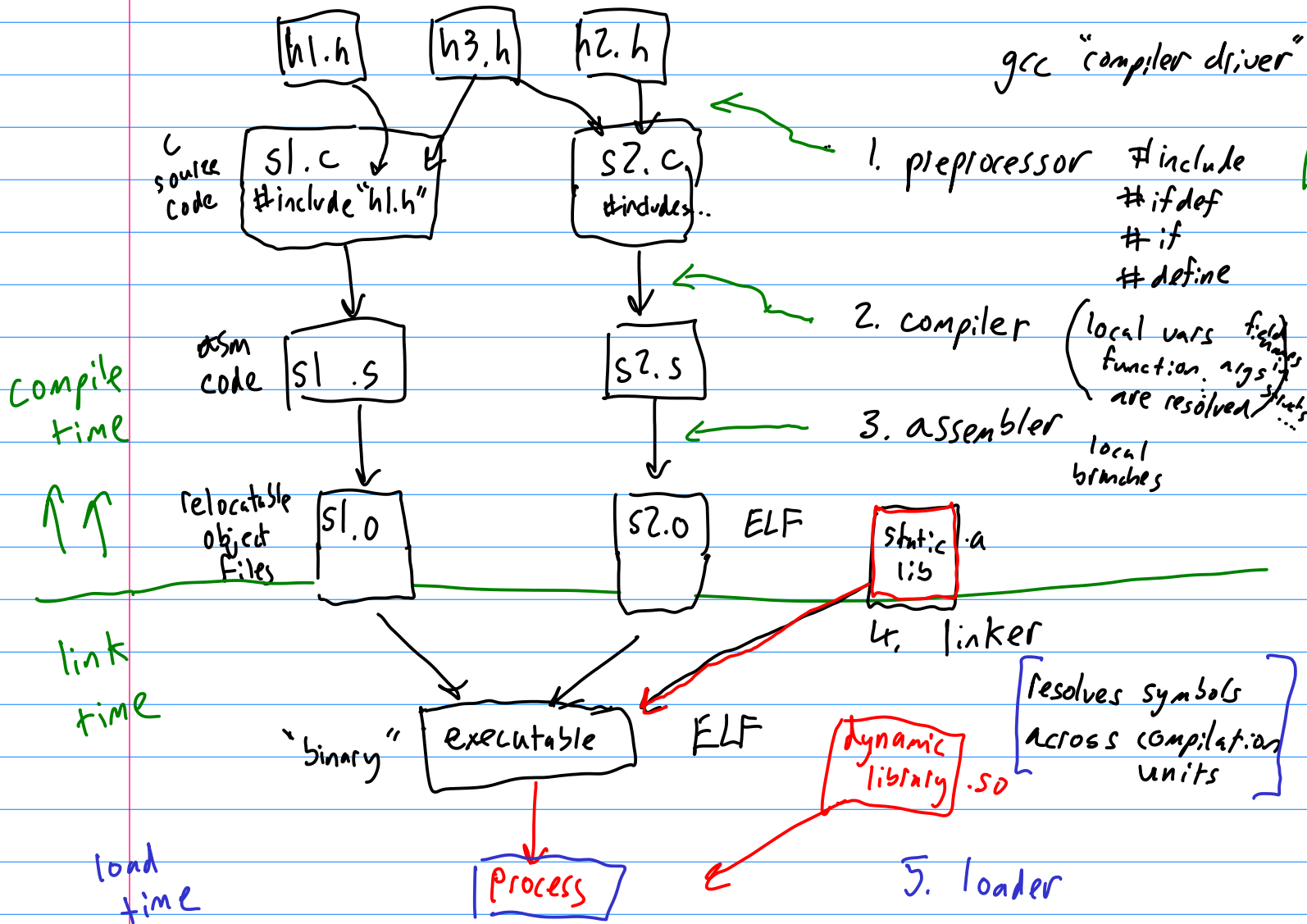
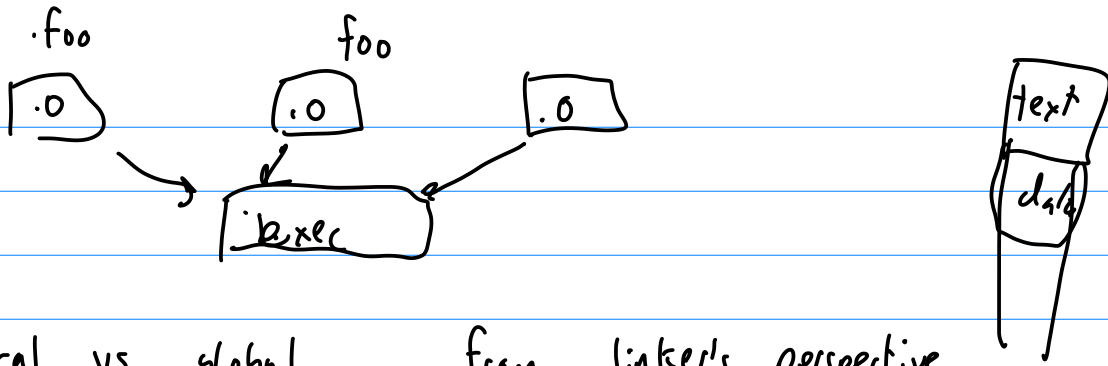


# CS 3214 lecture # 11 "more on linking"





local vs. global from linker's perspective

static keyword tells linker symbol local

```
static int v;
static int foo() { };
```

strong vs. weak symbols

strong + strong = conflict

strong + weak = ignores weak

weak + weak = ignore one use one

} hard to predict

when is a symbol strong vs. weak?

declarations vs. definitions in C

functions

```
(extern) int f(void);
```

← declaration (nothing is actually defined)

```
local → static int f() { };
```

← definition

↑ strong symbol

variables

```
int v;
```

← definition set to 0

↑ weak symbol

```
int v = 10;
```

← definition set to 10

↑ strong symbol

bsj

← declaration  
extern int v;

what if? in header file

funcs

{ extern void f();  
void f(){}  
static void f(){} }

← how you specify fun

X strong symbol conflict

← usually ok for inlining

vars

{ next time