

# Linking - Rules for symbol resolution

local symbols → only used in local module

```
int f() {
```

```
    static int x = 0;
```

```
}
```

→ local symbol

↳ in ~~global~~  
global section  
.data

```
int g() {
```

```
    static int x = 12;
```

```
}
```

x is NOT

a local var

and is NOT

on the stack

Global symbols → can be used from any module

module

```
int x = 0;
```

```
int f() {
```

```
    x++;
```

```
}
```

→ global symbol

mod2.c

```
g() {
```

```
    x x++;
```

```
}
```

functions also global symbols

Global symbols can be strong or weak

strong → functions + initialized variables

weak → uninitialized global variables

```
mod 1.c strong
int y = 5;
int x = 123;
void f();
void f() {
    int main() {
        f();
        return 0;
    }
}

mod 2.c weak
int x; int x = 57;
void f() { double y;
    x++;
    y++ = 7.5;
}
```

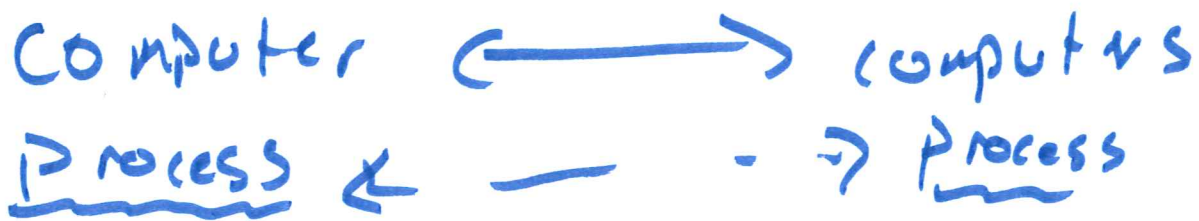
3 rules for symbol resolution

- 1) multiple strong symbols not allowed
- 2) given a strong + a weak symbol choose strong
- 3) given 2 weak symbols, choose any

# Networking!

- client/server programming model
  - Networking H/W
  - OSI model
  - Internet protocols (DNS)
  - Socket programming
- } Leo next week

Why networking?



Client/server programming model

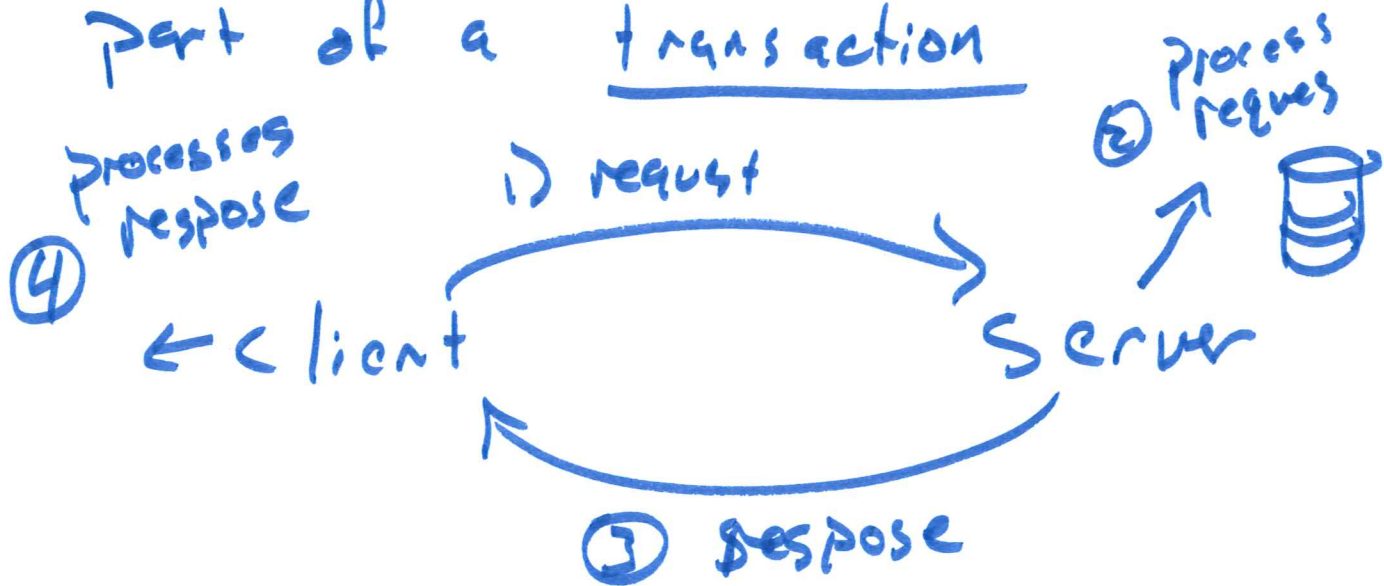
Server  $\rightarrow$  has a service

manages a resource

Client sends requests to server

Server sends responses to client

Part of a transaction



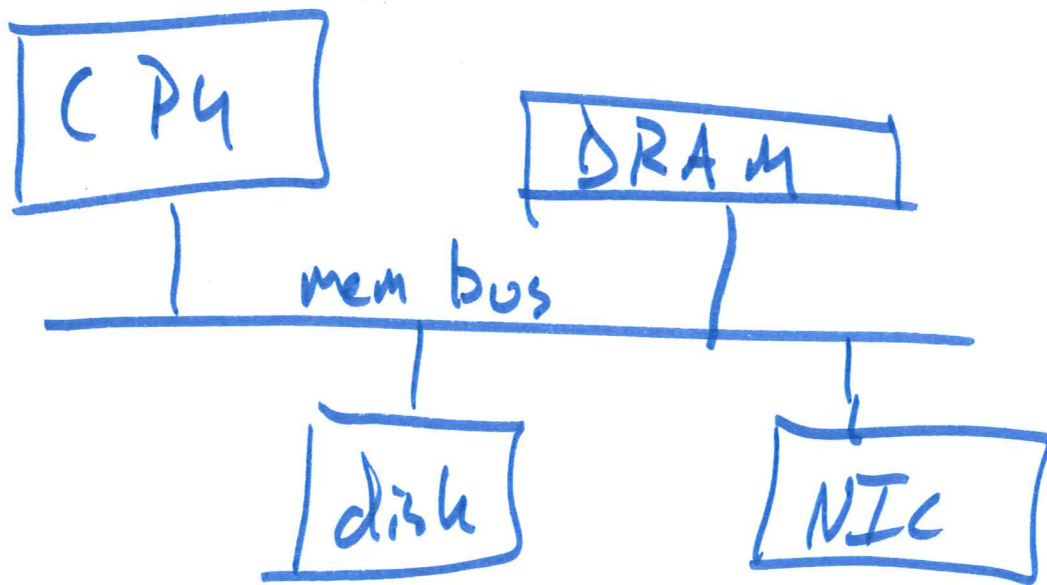
## "The Cloud"

To talk to the network

each system has a

Network Interface Controller (NIC)  
Network adapter Card





NIC just another I/O device

- like disk it uses MMIO + DMA
- converts electrical signals to 1's + 0's
- copies contents ~~of~~ via DMA to memory
- All MMIO ops are done by OS
  - many processes share I/O devices

→ Examples?

- cable
- fiber
- DSL
- Infiniband

- ethernet - 46 + 36
- wifi
- bluetooth
- serial
- telephones