Linking - Rules for symbol resolution

local symbols \(\rightarrow\) only used in local module

\[
\text{int } f() \equiv \quad \text{local symbol}
\]

\[
\text{Static int } x = 0; \quad \text{by in } \text{global section data}
\]

\[
\text{int } g() \equiv x \quad \text{is NOT a local var}
\]

\[
\text{Static int } x = 12; \quad \text{and is NOT on the stack}
\]

Global symbols \(\rightarrow\) can be used from any module

\[
\text{module} \quad \text{int } x = 0; \quad \text{global symbol} \quad \text{mod2.c}
\]

\[
\text{int } f() \equiv x++;
\]

\[
\text{functions also global symbols}
\]
Global symbols can be **strong** or **weak**

- **strong**: functions + initialized variables
- **weak**: uninitialized global variables

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

```
model 2. c  weak
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

Why networking?

Computer <--- Computer

Process <--- Process

Client/server programming model

Server -> has a service

manages a resource
Client sends requests to server
Server sends responses to client
Part of a transaction

1) Request
2) Process request
3) Response

"The Cloud"
To talk to the network
Each system has a
Network Interface Controller (NIC)
Network adapter card
NIC just another I/O device
- behave like disk, it uses MMIO + DMA
- converts electrical signals to 0's and 1's
- copies contents via DMA to memory
- All MMIO ops are done by OS
- Many processes share I/O devices

Examples:
- ethernet, serial, telephones
- cable, fiber, Wi-Fi, Bluetooth, Infiniband, DSL