

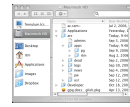
L3: Naming systems

6.033 Spring 2008
<http://web.mit.edu/6.033>

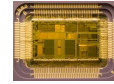


Fundamental abstractions

- Memory
 - Read/write
- Interpreter
 - Instruction repertoire
 - Environment
 - Instruction pointer
- Communication links
 - Send/receive



(loop (print (eval (read))))

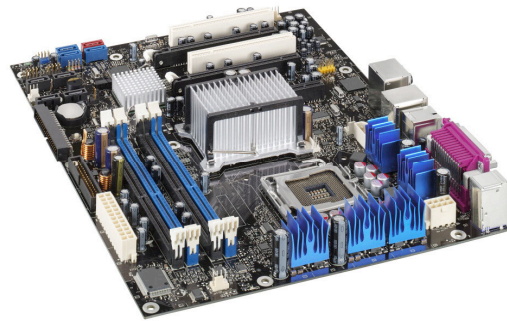


Web names

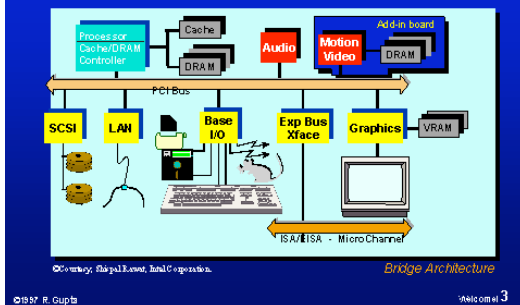
<http://web.mit.edu/6.033>

<http://www.google.com/search?q=6.033&start=0&ie=utf-8&oe=utf-8&client=firefox-a&rls=org.mozilla:en-US:official>

PC board



Anatomy Of A Personal Computer



Abstract bus picture

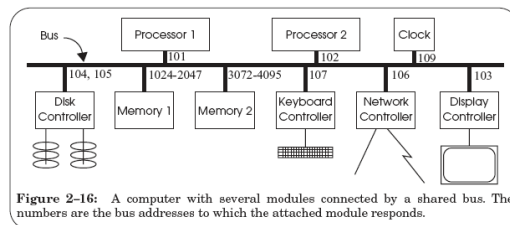


Figure 2-16: A computer with several modules connected by a shared bus. The numbers are the bus addresses to which the attached module responds.

- address is overloaded name with location info
- LOAD 1742, R1

Names

1. LOAD 1742, R1
2. 18.7.22.69
3. web.mit.edu
4. http://tinyurl.com/23a9dm
5. 6.033-staff@mit.edu
6. 203203021606.g24G64795174@scs.cs.nyu.edu
7. amsterdam
8. /mit/6.033/www
9. wc

Example: program linking

```

foo (...) {
  int x, y;
  init()
  x = sqrt(...)
  y = sqrt(x)
  printf ("answer: %d", y)
}

static init() { ...}

0xff00: // foo:
...
      jmp 0xff20 // init
...
      jmp 0xffd0 // sqrt
...
      jmp 0xffd0 // sqrt
...
      jmp 0xdc00 // printf

```

Step 1: compilation

```

foo (...) {
  int x, y;
  init();
  x = sqrt(...);
  y = sqrt(x);
  printf ("answer: %d", y);
}

static init() { ...}

gcc -c foo.c

foo.o:
• Text:
  0x00: // foo:
    jmp +20 // init()
  ...
  jmp ??? // sqrt
  ...
• Symbol table:
  [foo, 0]
• Relocation tbl:
  [sqrt, 12]
  [sqrt, 20]
  [printf, 30]

```

Step 2: linking

- gcc main.o foo.o /usr/lib/libm.a /usr/lib/libc.a

```

foo.o:
• Text:
  0x00: // foo:
  ...
  jmp ??? // printf
  ...
• Symbol table:
  [foo, 0]
• Relocation tbl:
  [sqrt, 12]
  [sqrt, 20]
  [printf, 30]

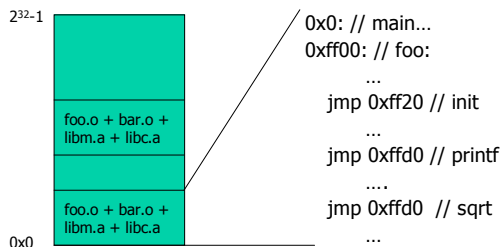
+

libc.a (printf.o, malloc.o):
printf.o:
• Text:
  0x00: // printf:
  jmp +20 // init()
  ...
  jmp ??? // malloc()
• Symbol table:
  [printf, 0]
• Relocation table:
  [malloc, 18]

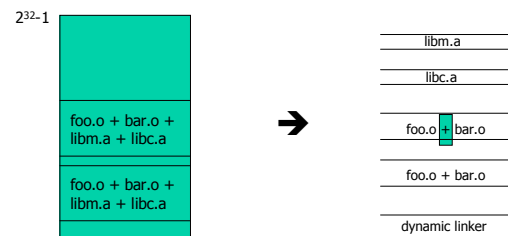
```

Step 3: loading

- Load in memory:



Sophistication: dynamic linking



- Resolve names at runtime through indirection

Summary

- Understanding a naming system:
 - What is the syntax for name?
 - What are the values?
 - What is the naming resolution algorithm?
 - Where does a name's context from?