Lecture 18

Andrea LaPaugh

April 11, 1996

(see Lecture 17 notes for much of lecture)

A linking example

/u/cs126/toy/powers.toy revisited

Subroutine to compute powers

R1: x R2: exponent

R3: return value R5: return address

assume R0 always contains 0

```
00: B401    R4 <- 1   use as constant
01: B301    R3 <- 1    initialization
02: 2224    R2 <- R2-R4 = R2-1
03: 6206    if R2 < 0 jump to end
04: 3331    R3 <- R3*R1
05: 5002    jump to top of loop
06: 7500    jump to address in R5
```

To compute R6 <- R6 ^4 + R7 ^5

```
00: B100    R0 <- 0
01: 1160    R1 <- R6+R0 = R6
02: B204    R2 <- 4
03: 85??    call subroutine,  R5 holds return
04: 1630    R6 <- R3+R0 = R3
05: 1170    R1 <- R7+R0 = R7
06: B205    R2 <- 5
07: 85??    call subroutine,  R5 holds return
08: 1663    R6 <- R6 + R3
09: 4602    write R6 to tty
0A: 0000    halt
```

Other OS tasks

A primary function of modern operating systems is to provide simple mechanisms for

* saving/accessing data (file system)
* user interaction (window system)

File System

- relatively old
- basic ideas required on ancient systems
- organize disk and provide table of where to find files

Window Manager

- relatively new
- history
  - Xerox PARC (Alto)
  - Macintosh
  - Windows NT
  - X
- each window a "virtual" terminal

Using files and windows

We won't have time to look at details of these

Use file system all the time

- new assignment (7):
  - get to manipulate more directly

Use window system all the time

- may have assignment where
  - manipulate more directly (X toolkit)