Assembly Programming

Professor Ming-Jer Tsai 蔡明哲
Department of Computer Science
Outline

- Goal
- 80x86 Assembly
- INT 21H
- INT 16H
Goal

- Use DOS DEBUG to compute $n!$, where $n$ is input from keyboard
Register

- General registers (4): ax, bx, cx, dx.
- Pointer registers (4): sp, bp, si, di, ip.
- Segment registers (4): cs, ds, es, ss.
- Flag register (1): af, cf, of, sf, pf, zf, df, if, tf.
**Move**

- `mov r1/m1, r2/m2/data`
  - Move `r2/m2/data` to `r1/m1`.
  - `r1` and `r2` are not allowed to be segment registers at a time.
  - `m1` and `m2` are not allowed at a time.
  - Only general register data can be moved to segment register.

- Flag register is not affected.

- Example:
  - `mov ax, bx`
  - `mov ds, cs` (illegal)
  - `mov cs, m` (illegal)
Arithmetic

- **add/adc r1/m1, r2/m2/data**
  - “add” move r1/m1+r2/m2/data (result) to r1/m1.
  - “adc” move r1/m1+r2/m2/data+C (result) to r1/m1.
  - m1 and m2 are not allowed at a time.
  - of is set if the result > 7fff.
  - cf is set if the result > ffff.
Arithmetic (2)

- sub/sbb r1/m1, r2/m2/data
  - “sub” move r1/m1 - r2/m2/data (result) to r1/m1.
  - “sbb” move r1/m1 - r2/m2/data - C (result) to r1/m1.
- m1 and m2 are not allowed at a time.
- cf is set if the result > 7fff.
- sf is set if the result > 7fff.
Arithmetic (3)

- inc/dec/neg r1/m1
  - “inc” move r1/m1+1 to r1/m1.
  - “dec” move r1/m1-1 to r1/m1.
  - “neg” move 0-r1/m1 to r1/m1.
  - “neg” set of if result = -32768.
  - “neg” reset cf if r1/m1 = 0 and set cf otherwise.
Arithmetic (4)

- `cmp r1/m1, r2/m2/data`
- `m1` and `m2` are not allowed at a time.
- Flag register is affected by the result equal to `sub r1/m1, r2/m2/data`
Arithmetic (5)

- **mul r1/m1**
  - “mul” can be byte and word operator.
- **Example:**
  - If mul bl, then ax=al*bl.
  - If mul bx, then (dx, ax)=ax*bx
Logic

- and/or r1/m1,r2/m2/data
- “and” move r1/m1&r2/m2/data to r1/m1
- “or” move r1/m1|r2/m2/data to r1/m1
- m1 and m2 are not allowed at a time.
Branch

- Loop label
  - If $cx \neq 0$, then jump to label and dec $cx$.
- jmp r1
- jmp/jxx m1
  - $xx=c, cxz, e, g, ge, l, le, o, p, s, z, nc, ne, ng, nge, nl, nle, no, np, ns, nz$.
  - $c$ (carry): jump if $cf=1$; $cxz$: jump if register $cx=0$; $e$ (equal): jump if $zf=1$; $g$ (greater): jump if $zf=0$ and $sf=0$; $l$ (less): jump if $sf \neq 0$; $n$ (not); $o$ (overflow): jump if $of=1$; $pf$ (parity): jump if $pf=1$; $s$ (sign): jump if $sf=1$; $z$ (zero): jump if $zf=1$;
INT 21H

- Terminate program and return DOS
  - ah=4ch

- Example:
  - mov ah, 4ch
  - int 21h
INT 16H

- Read one character from keyboard
  - ah=00h
  - al=ASCII code of the input character

Example:
- mov ah, 00h
- int 16h