



การจัดองค์การคอมพิวเตอร์

พ4.6 ภาษาแอสเซมบลี (3/3)

31110321 Computer Organization

สำหรับนักศึกษาชั้นปีที่ 3 สาขาวิชาวิศวกรรมคอมพิวเตอร์

ทรงฤทธิ์ กิตติศรีวรพันธุ์

songrit@npu.ac.th

สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยนครพนม

Lecture plan

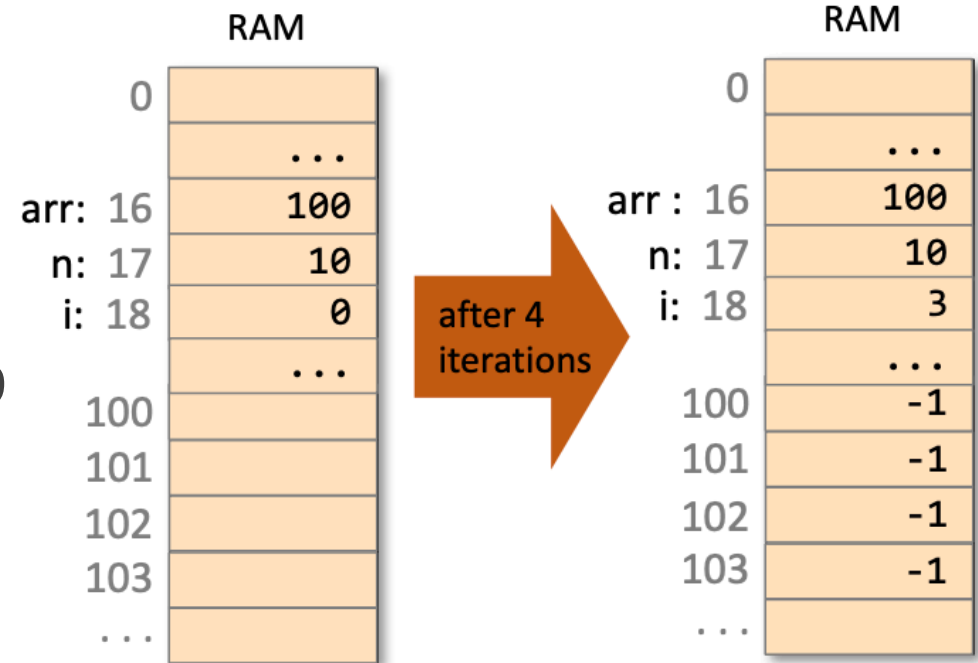
- 4.1 ภาษาเครื่อง
- 4.2 ส่วนประกอบพื้นฐาน
- 4.3 ระบบแอสกัคคอมพิวเตอร์และภาษาเครื่อง
- 4.4 ภาษาเครื่องแอสกัค
- 4.5 อินพุต / เอาท์พุต
- **4.6 การเขียนโปรแกรมสำหรับเครื่องแอสกัค (3/3)**
- 4.7 ภาพรวมโปรเจกต์สัปดาห์ 4

พอยต์เตอร์

Example:

```
// for (i=0; i<n; i++) {  
//     arr[i] = -1  
// }
```

- อาร์เรย์ เป็นบุลล็อกบนหน่วยความจำ
- เมื่อต้องการอ่านอาร์เรย์เราต้องอยู่ตำแหน่งหน่วยความจำนั้น
- เราต้องการตัวแปร บุนทีกตำแหน่งหน่วยความจำ
- เรียกตัวแปรนั้นว่า “พอยต์เตอร์”



พอยต์เตอร์

```
// for (i=0; i<n; i++) {  
//     arr[i] = -1  
// }  
  
// Suppose that arr=100 and n=10  
  
// Let arr = 100  
@100  
D=A  
@arr  
M=D  
  
// Let n = 10  
@10  
D=A  
@n  
M=D  
  
// Let i = 0  
@i  
M=0  
  
// Loop code continues  
// in next slide...
```

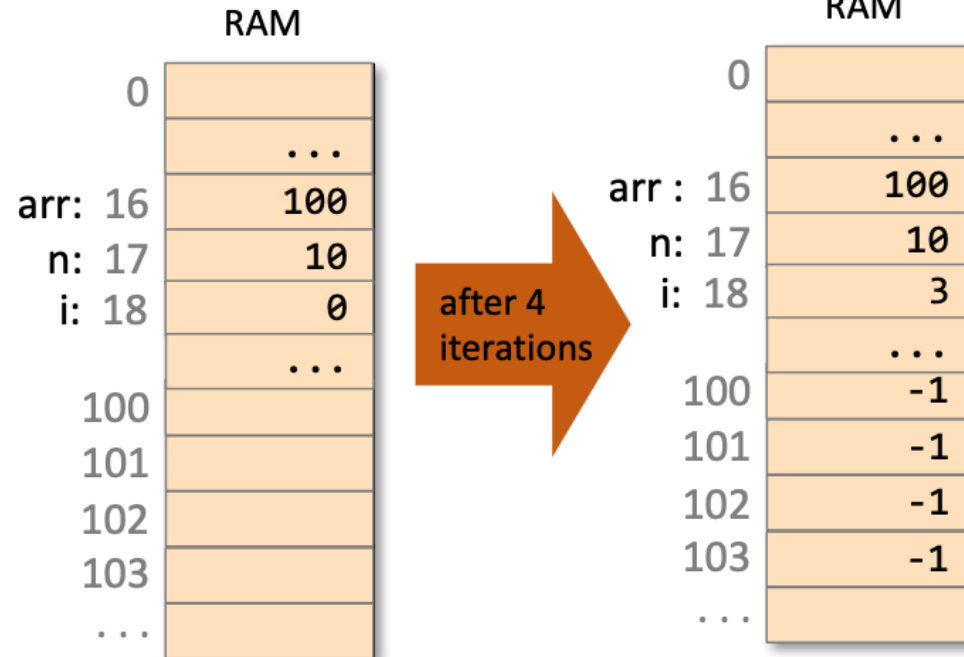
Nand to Tetris / www.nand2tetris.org / Chapter 4 / Con



พอยต์เตอร์

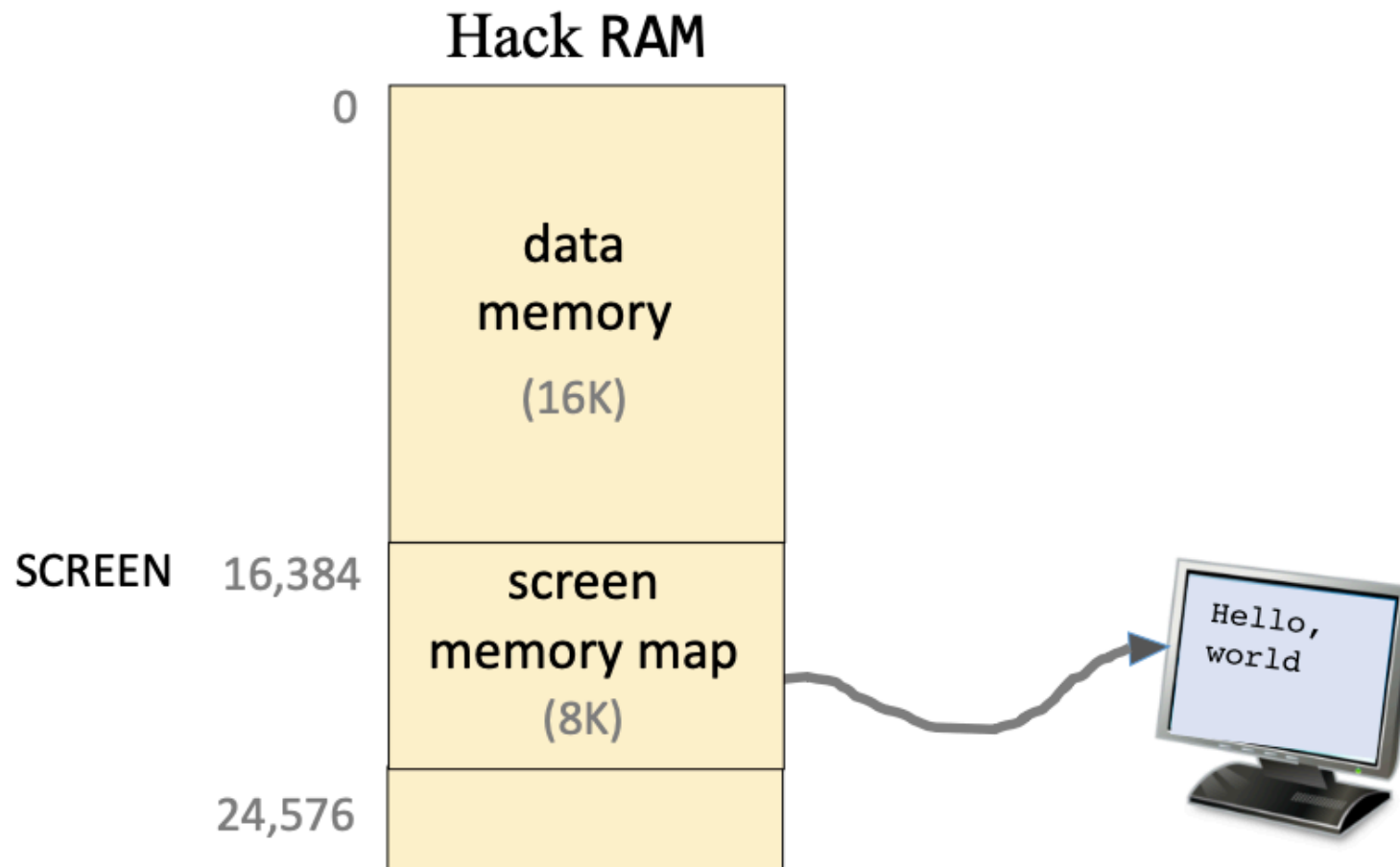
```
(LOOP)
// if (i==n) goto END
@i
D=M
@n
D=D-M
@END
D;JEQ

// RAM[arr+i] = -1
@arr
D=M
@i
A=D+M
M=-1
```



Screen

- ใช้ Screen เริ่มที่ 16,384



Screen

The screenshot shows a debugger interface with a menu bar (File, View, Run, Help) and a toolbar. The main area is divided into three panes: ROM, RAM, and a Screen window.

ROM Pane: Displays assembly code. Address 27 is highlighted in yellow.

Address	Code
0	@0
1	D=M
2	@16
3	M=D
4	@17
5	M=0
6	@16384
7	D=A
8	@18
9	M=D
10	@17
11	D=M
12	@16
13	D=D-M
14	@27
15	D;JGT
16	@18
17	A=M
18	M=-1
19	@17
20	M=M+1
21	@32
22	D=A
23	@18
24	M=D+M
25	@10
26	0;JMP
27	@27
28	0;JMP

RAM Pane: Displays memory values. Address 16 is highlighted in yellow.

Address	Value
0	50
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	50
17	51
18	18016
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0

Screen Window: A large white area representing the screen. A black rectangle is drawn in the upper left corner. Annotations indicate it is "50 pixels long" and "16 pixels wide". An orange callout bubble labeled "Screen" points to the window.

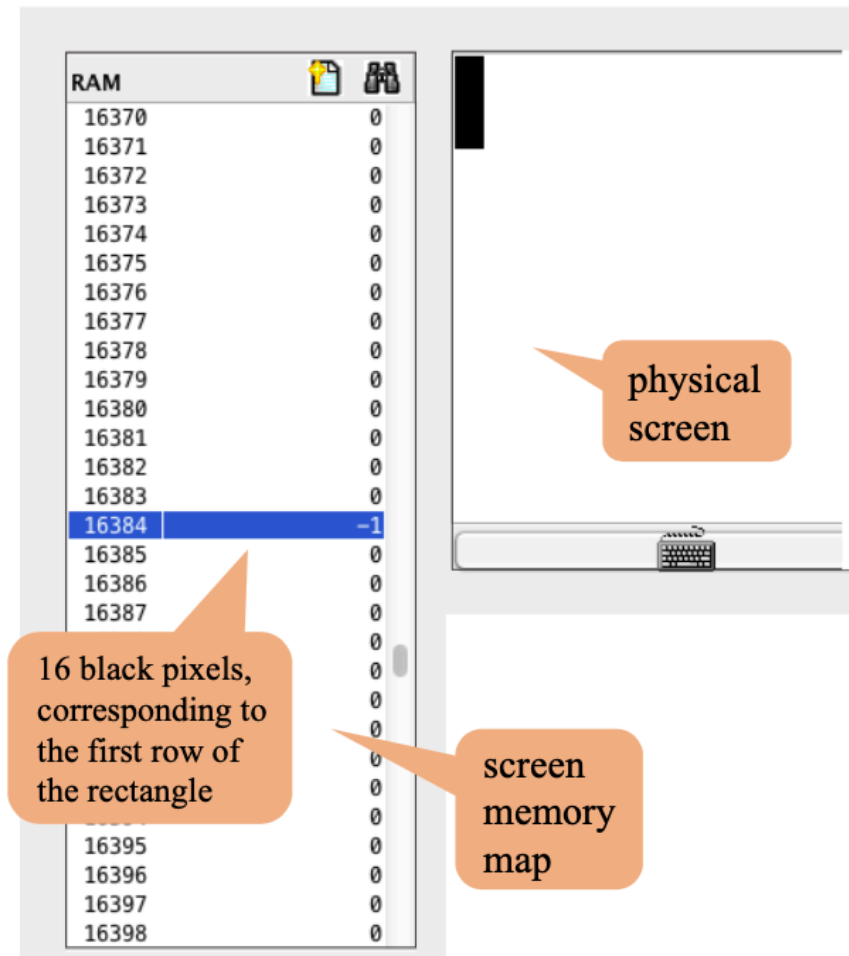
Task: draw a filled rectangle at the upper left corner of the screen, 16 pixels wide and RAM[0] pixels long

Registers: At the bottom, the PC register is 27 and the A register is 27.

ควบคุมการแสดงผล

- Pseudo code

```
// for (i=0; i<n; i++) {  
//     draw 16 black pixels at the  
//     beginning of row i  
// }  
  
addr = SCREEN  
n = RAM[0]  
i = 0  
  
LOOP:  
    if i > n goto END  
    RAM[addr] = -1 // 1111111111111111  
    // advances to the next row  
    addr = addr + 32  
    i = i + 1  
    goto LOOP  
  
END:  
    goto END
```



ควบคุมการแสดงผล

Assembly code

```
// Program: Rectangle.asm
// Draws a filled rectangle at the
// screen's top left corner, with
// width of 16 pixels and height of
// RAM[0] pixels.
// Usage: put a non-negative number
// (rectangle's height) in RAM[0].
```

```
@SCREEN
```

```
D=A
```

```
@addr
```

```
M=D // addr = 16384
      // (screen's base address)
```

```
@0
```

```
D=M
```

```
@n
```

```
M=D // n = RAM[0]
```

```
@i
```

```
M=0 // i = 0
```

(continued)

```
(LOOP)
```

```
@i
```

```
D=M
```

```
@n
```

```
D=D-M
```

```
@END
```

```
D;JGT // if i>n goto END
```

```
@addr
```

```
A=M
```

```
M=-1 // RAM[addr]=1111111111111111
```

```
@i
```

```
M=M+1 // i = i + 1
```

```
@32
```

```
D=A
```

```
@addr
```

```
M=D+M // addr = addr + 32
```

```
@LOOP
```

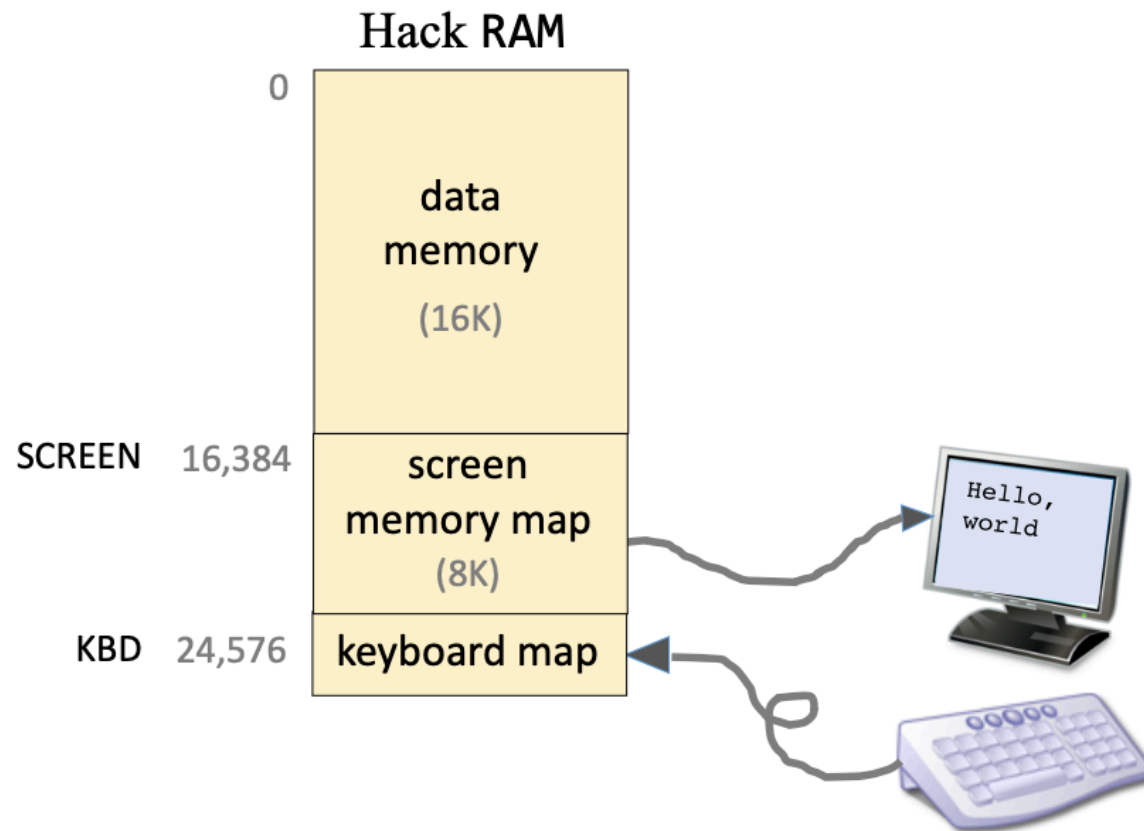
```
0;JMP // goto LOOP
```

```
(END)
```

```
@END // program's end
```

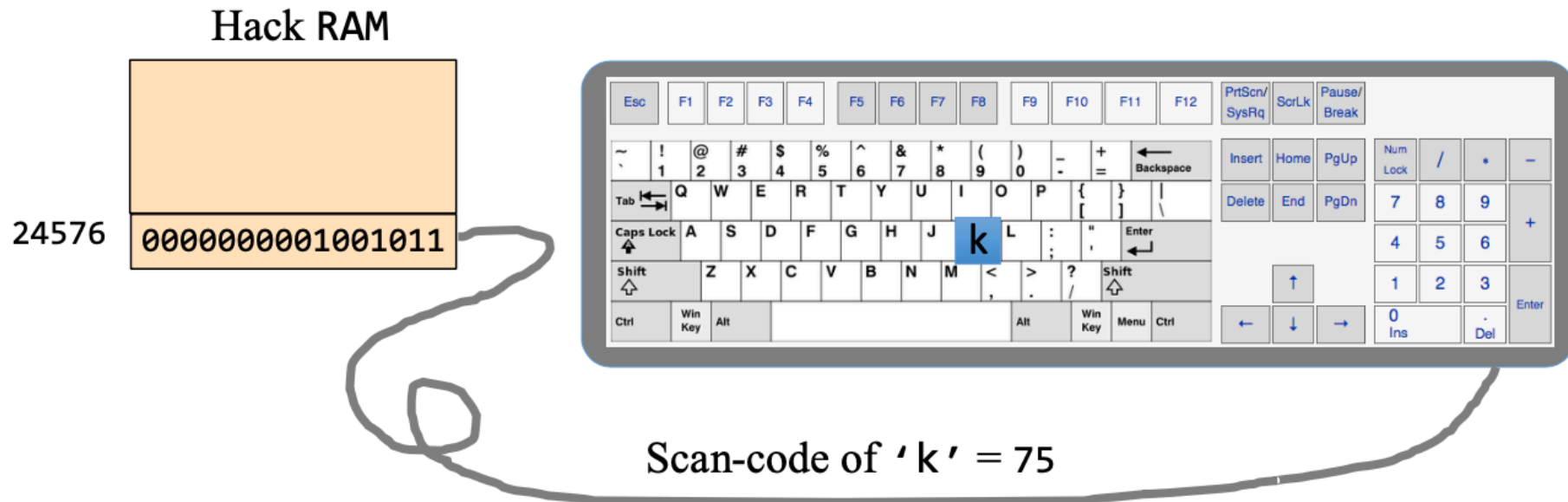
```
0;JMP // infinite loop
```

KBD : อินพุต



- SCREEN เก็บหน่วยความจำจอภาพ
- KBD เก็บตำแหน่งหน่วยความจำของคีย์บอร์ด

Handling the keyboard



- อ่านข้อมูลในตำแหน่ง RAM[24576] (@KBD)
- ถ้าไม่มีกดปุ่ม ข้อมูลในหน่วยความจำเป็น 0, หมายถึงไม่กดปุ่ม
- ถ้ามีกดปุ่ม ให้แสดงผลตามรหัสคีย์บอร์ด

ปิดท้าย

High level code

```
for (i=0; i<n; i++) {  
    arr[i] = -1  
}
```

Compiler

Machine language

```
...  
@i  
M=0  
(LOOP)  
@i  
D=M  
@n  
D=D-M  
@END  
D;JEQ  
@arr  
D=M  
@i  
A=D+M  
M=-1  
@i  
M=M+1  
@LOOP  
0;JMP  
(END)  
@END  
0;JMP
```

- แอสเซมบลีเป็นภาษาระดับล่าง
- Profound
- Subtle
- Efficient (or not)
- Intellectually challenging

Lecture plan

- 4.1 ภาษาเครื่อง
- 4.2 ส่วนประกอบพื้นฐาน
- 4.3 ระบบแอสกัคคอมพิวเตอร์และภาษาเครื่อง
- 4.4 ภาษาเครื่องแอสกัค
- 4.5 อินพุต / เอาท์พุต
- 4.6 การเขียนโปรแกรมสำหรับเครื่องแอสกัค (3/3)
- **4.7 โปรเจ็ค**