บทที่ 1

ความรู้เบื้องต้นเกี่ยวกับโปรแกรมระบบ

การวัดและประเมินผล

- สอบกลางภาค (Midterm) 30
- สอบปลายภาค (Final) 30
- งาน Homework, Program 30
- เข้าเรียน+จิตพิสัย 10
- รวม 100

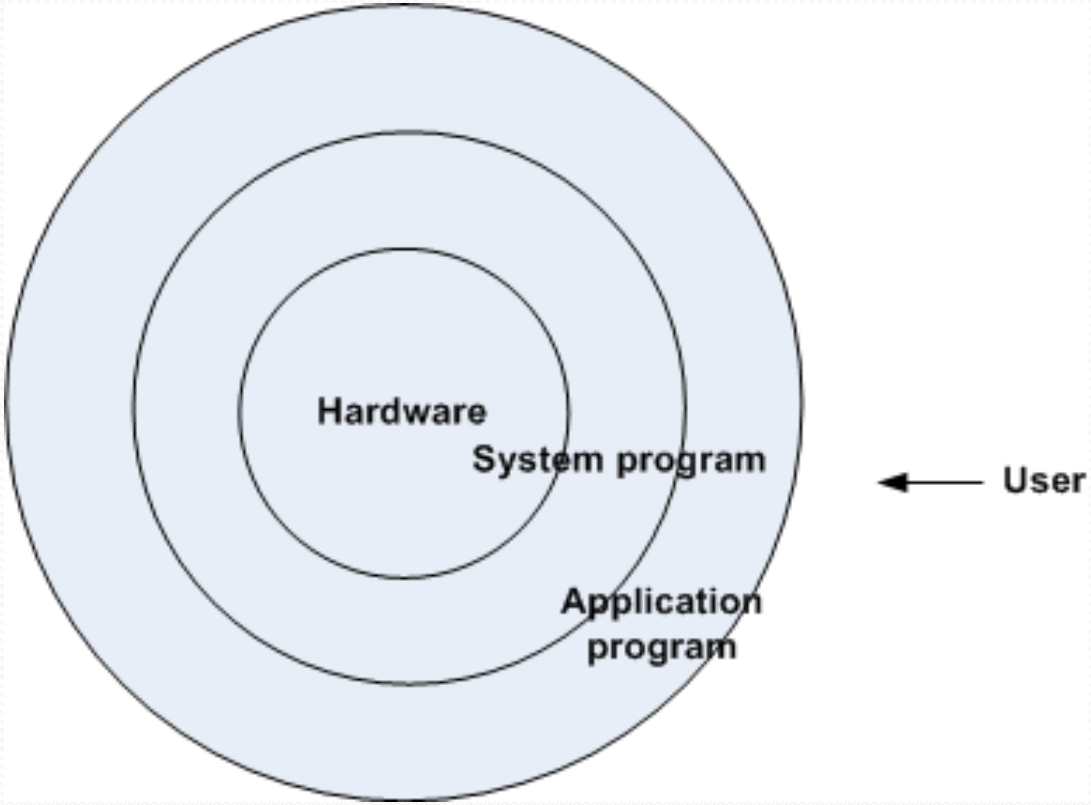
โปรแกรมระบบ (Systems Software) คืออะไร

- โปรแกรมที่ถูกพัฒนาขึ้น เพื่อช่วยให้ผู้ใช้คอมพิวเตอร์ (Users) สามารถใช้คอมพิวเตอร์ ได้ตรงตามความต้องการของผู้ใช้ ง่าย และสะดวก
- โปรแกรมที่ใช้ ช่วยในการสนับสนุนการทำงานของโปรแกรมประยุกต์

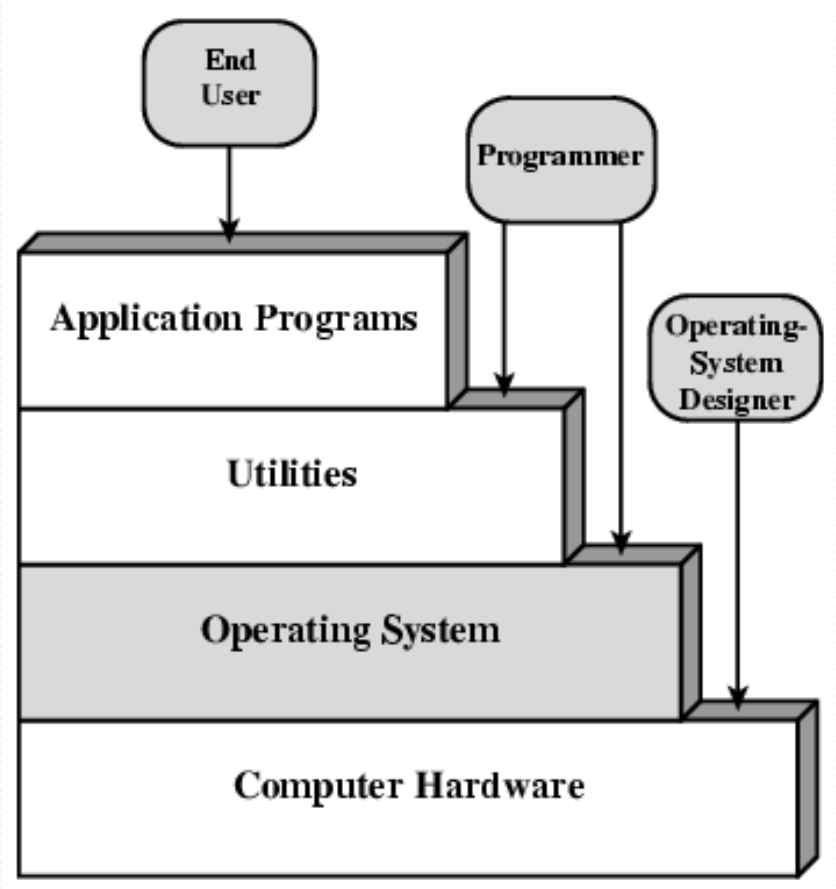
ตัวอย่างของโปรแกรมระบบ

- Assembler
- Compiler
- Macro Processor
- Loader
- Linker
- Operating System

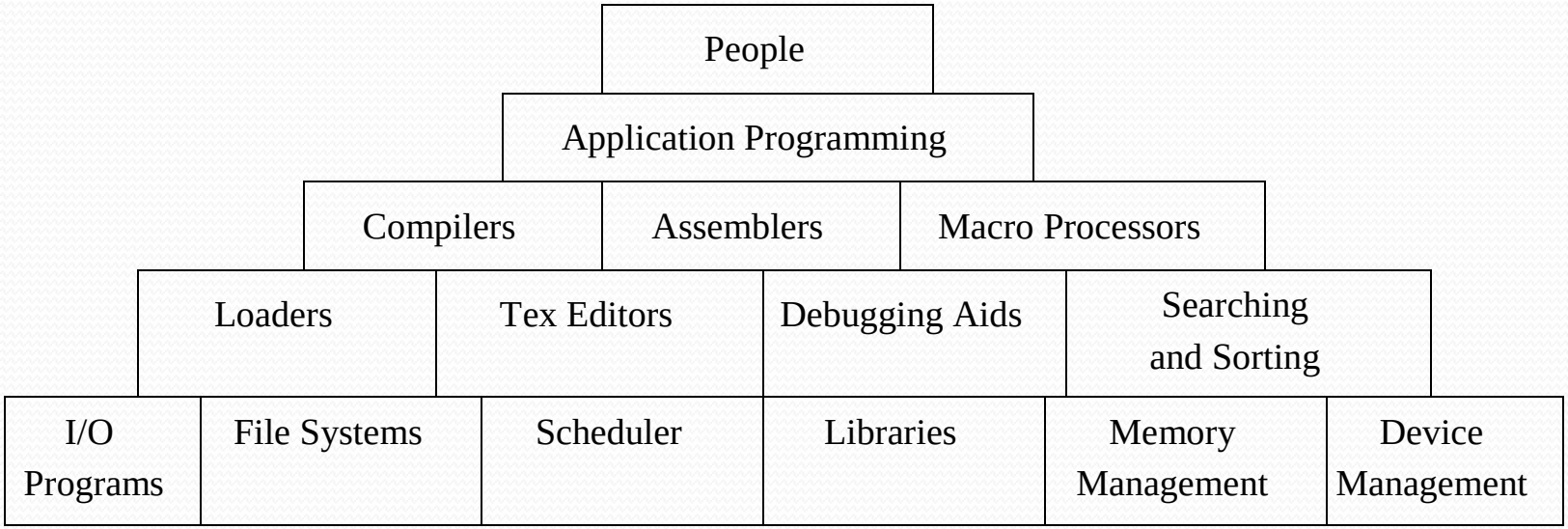
ลำดับชั้นของการใช้งานคอมพิวเตอร์



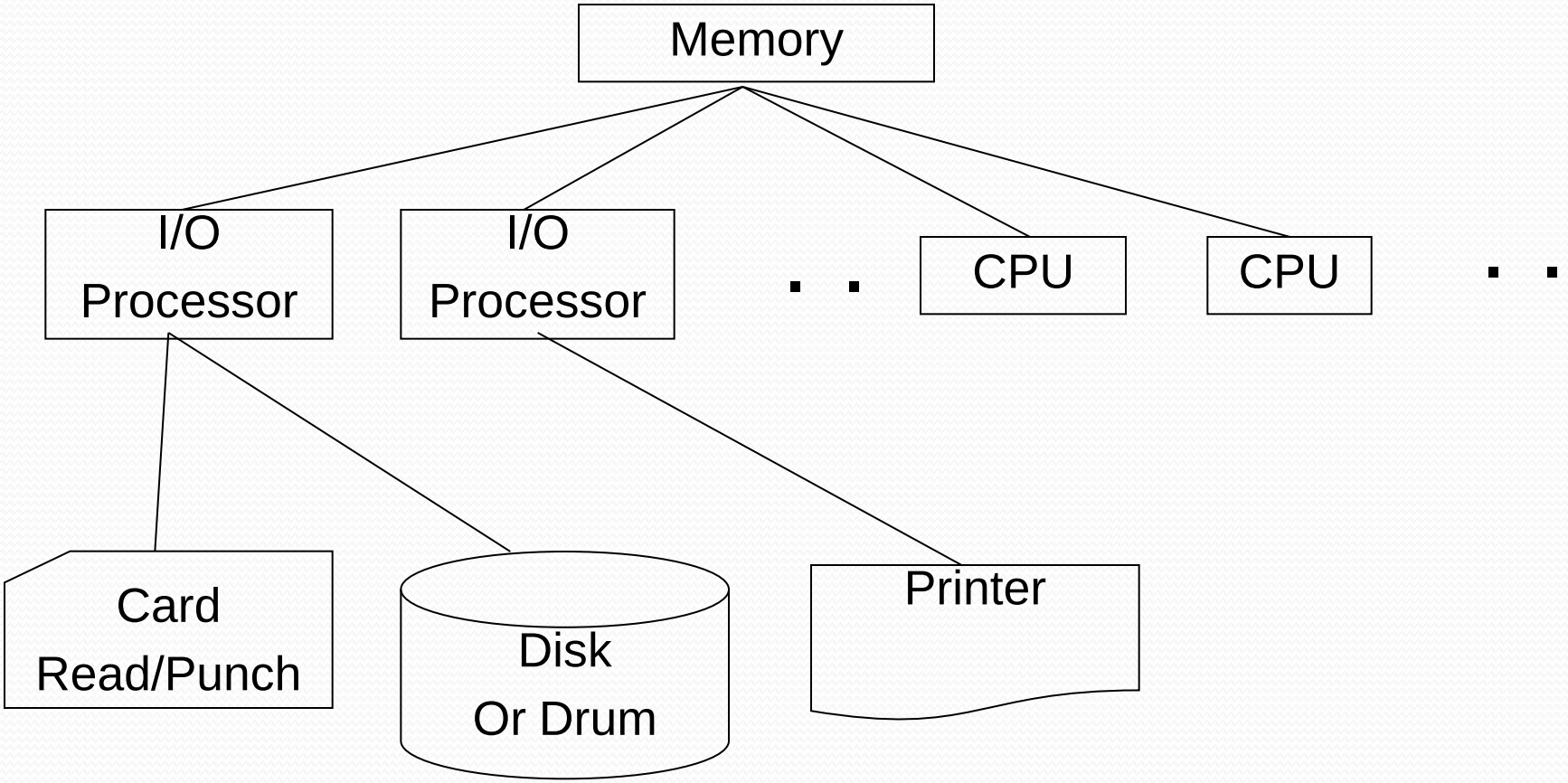
ลำดับชั้นของการใช้งานคอมพิวเตอร์



องค์ประกอบของการเขียนโปรแกรมระบบ



โครงสร้าง เครื่องคอมพิวเตอร์



โครงสร้าง เครื่องคอมพิวเตอร์ (ต่อ)

- Memory เป็นอุปกรณ์ที่ใช้เก็บข้อมูล และคำสั่ง
- Processor อุปกรณ์ที่ใช้ในการประมวลผลข้อมูล ตามคำสั่งที่เก็บไว้ในหน่วยความจำ
- ข้อมูล และคำสั่ง เก็บอยู่ในรูปของ ‘0’ และ ‘1’
- ‘0’ และ ‘1’ เรียก Binary Digits (Bit)
- เมื่อหลาย Bits มารวมกัน กลายเป็น Words, Characters หรือ Bytes เป็นต้น

โครงสร้าง เครื่องคอมพิวเตอร์ (ต่อ)

- Memory Location ถูกกำหนดโดย Address ที่ชี้ไปยังข้อมูล หรือคำสั่ง ในหน่วยความจำ
- Memory หรือหน่วยความจำ คล้ายกับ ตู้จดหมาย ที่ใช้บรรจุกลุ่มข้อมูล ‘0’ และ ‘1’ โดยมี Address กำกับ

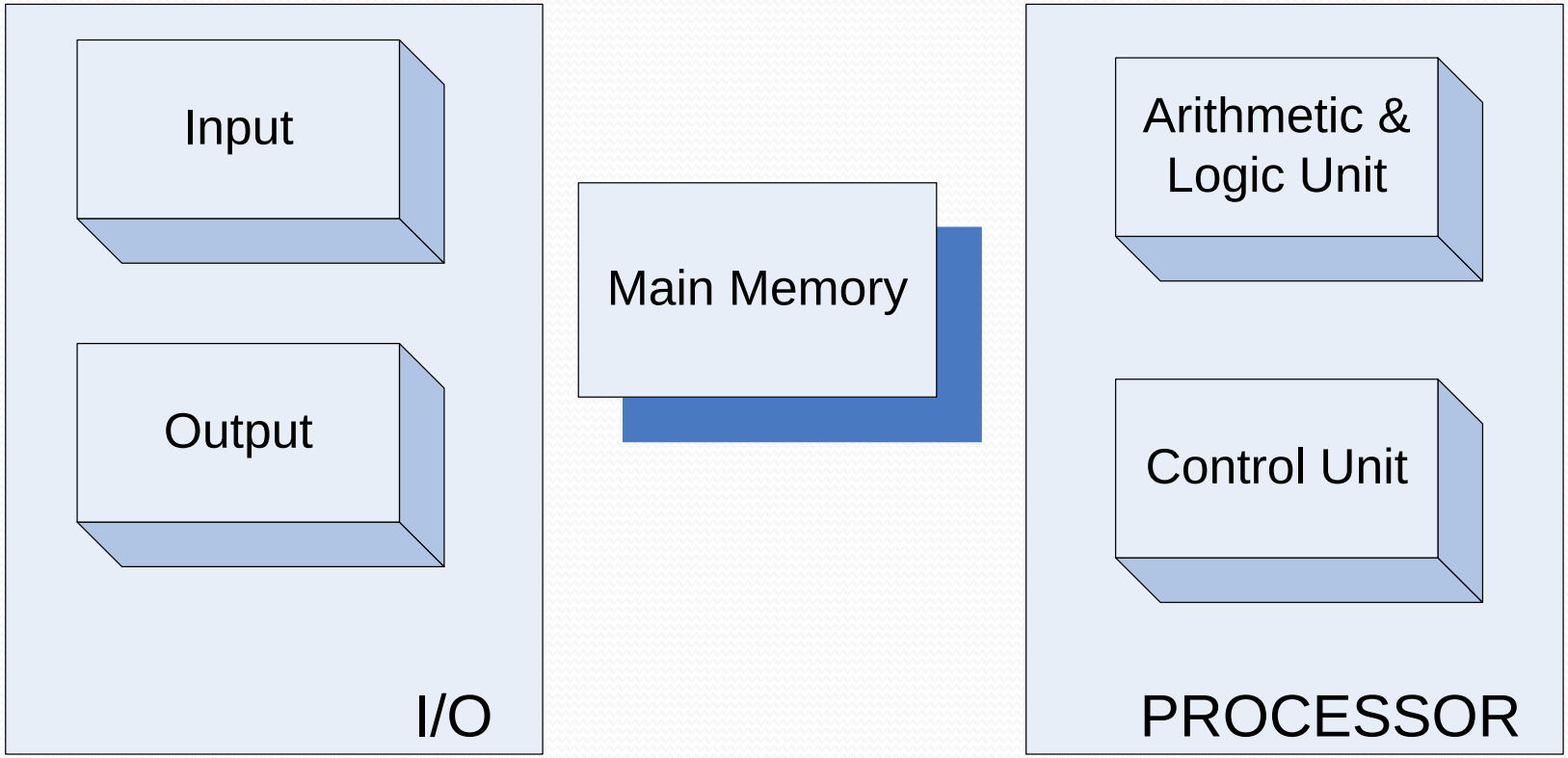
Address	Contents
10,000	0000 0000 0000 0001
10,001	0011 0000 0000 0001
10,002	0000 0000 0000 0101

โครงสร้าง เครื่องคอมพิวเตอร์ (ต่อ)

- Code หรือ รหัส คือกลุ่มของ Bits ที่เกิดจากกฎเกณฑ์ ที่เราตั้งขึ้น เช่น
  - BCD ใช้แทนตัวเลข (decimal digit)
  - EBCDIC และ ASCII ใช้แทนตัวอักขระ
  - Instruction ขึ้นอยู่กับตัว Processor

- Processor จะแบ่งเป็น 2 ประเภทใหญ่ ๆ คือ
  - CPU (Central Processing Unit) หน่วยประมวลผลกลาง
  - I/O processors

โครงสร้าง เครื่องคอมพิวเตอร์ (ต่อ)

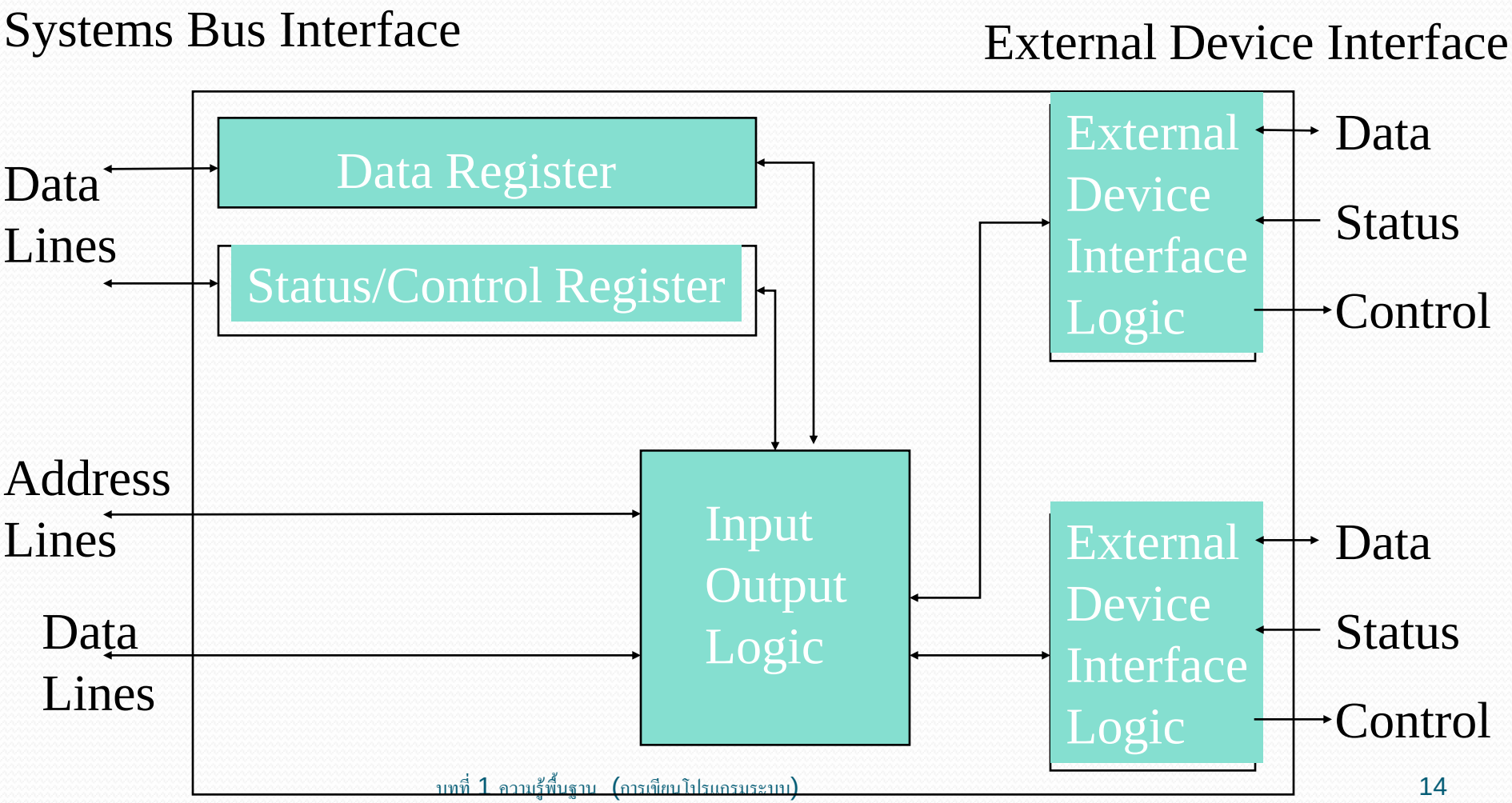


โครงสร้าง เครื่องคอมพิวเตอร์ (ต่อ)

- I/O Processor ทำหน้าที่ย้ายข้อมูลระหว่างหน่วยความจำ กับอุปกรณ์ประกอบภายนอก เช่น เครื่องพิมพ์ ดิสก์ เครื่องพิมพ์ เป็นต้น
  - ทำงานตามคำสั่งที่เก็บไว้ในหน่วยความจำ
  - สั่งให้ทำงานโดย CPU

- I/O Instructions คือกลุ่มคำสั่ง ที่ทำงานโดยตัว I/O Processors
  - ชุดคำสั่ง อาจแตกต่างจากชุดคำสั่งของ CPU
  - สามารถทำงานแบบ Asynchronously (Simultaneously) กับตัว CPU

โครงสร้าง เครื่องคอมพิวเตอร์ (ต่อ)



โครงสร้าง เครื่องคอมพิวเตอร์ (ต่อ)

- การทำงานแบบ Asynchronous ระหว่าง I/O และ CPU เป็นต้นแบบของ Multiprocessing

- Multiprocessing หมายถึง ระบบคอมพิวเตอร์ ที่มีตัว Processor มากกว่า 1 ตัว ทำงานพร้อมกัน โดยใช้หน่วยความจำ (Memory) ที่เดียวกัน

โครงสร้าง เครื่องคอมพิวเตอร์ (ต่อ)

- Procedure หรือโปรแกรมย่อย ที่แก้ไขตัวมันเอง (Modify) เรียกว่า Impure Procedure
  - Impure Procedure
    - อ่านเข้าใจยาก
    - ไม่สามารถทำงานร่วมกัน (Shared) แบบ Multiprocessor ได้
    - สั่งให้ทำงานแต่ละครั้ง (Execute) จะเจอคำสั่งต่างๆ กัน
    - เพราะฉะนั้น เอาโปรแกรม มาใช้หนหรือ 2 หนไม่ได้
  - ถ้าจะให้โปรแกรม Reusable เหมือนกันทุกครั้ง ที่ทำการ Execute ต้องเขียนโปรแกรมเป็น Pure Procedure หรือ Re-Entrant Code

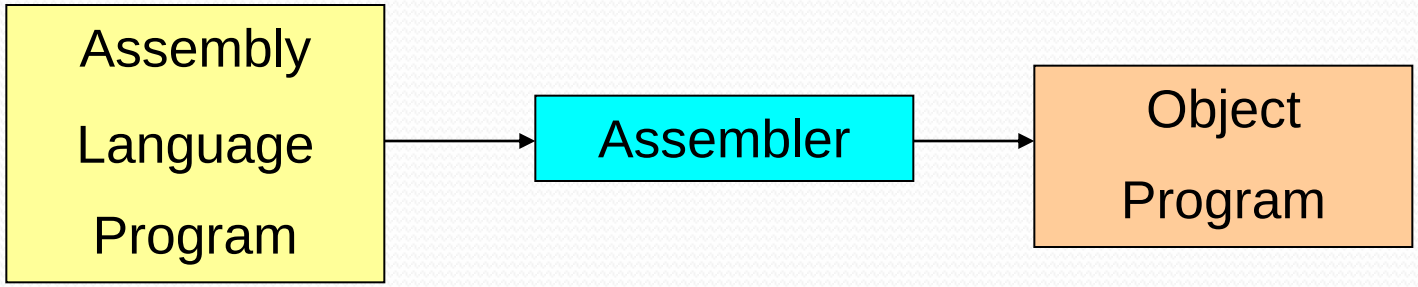
1.2 องค์ประกอบของโปรแกรมระบบ

1.2.1 Assembler

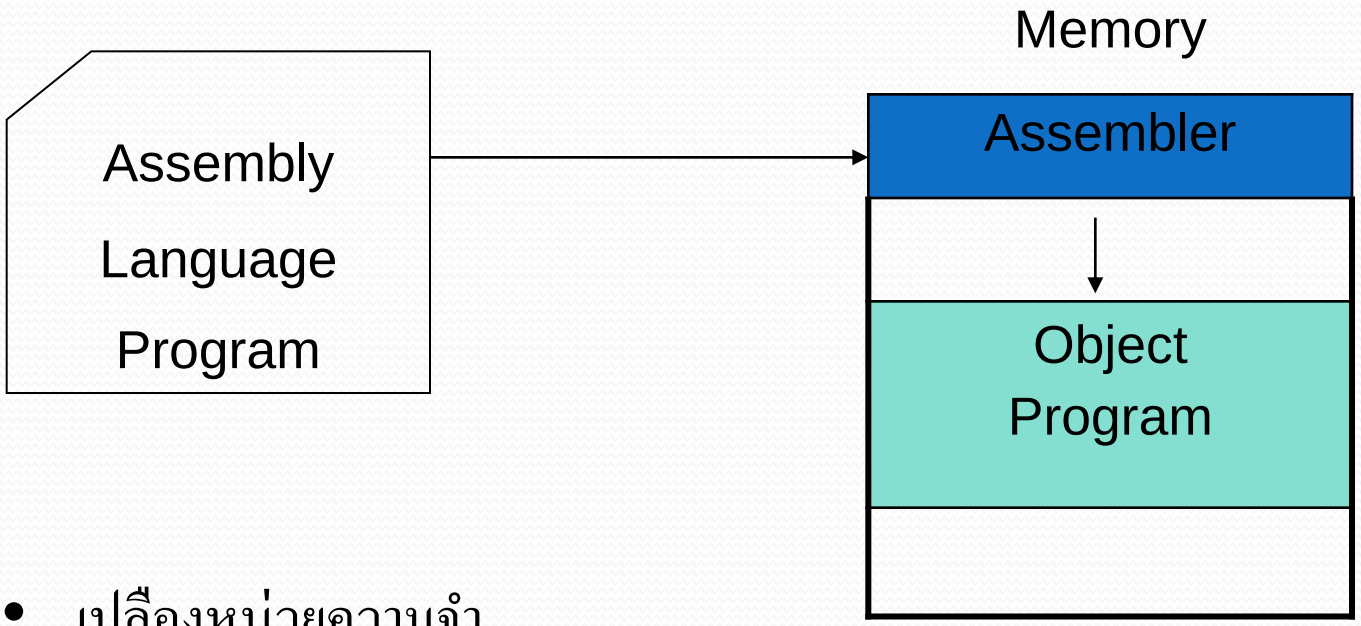
ใช้แปลงโปรแกรมภาษาแอสเซมบลี้ ให้เป็นภาษาเครื่อง

Source Program

Destination Program

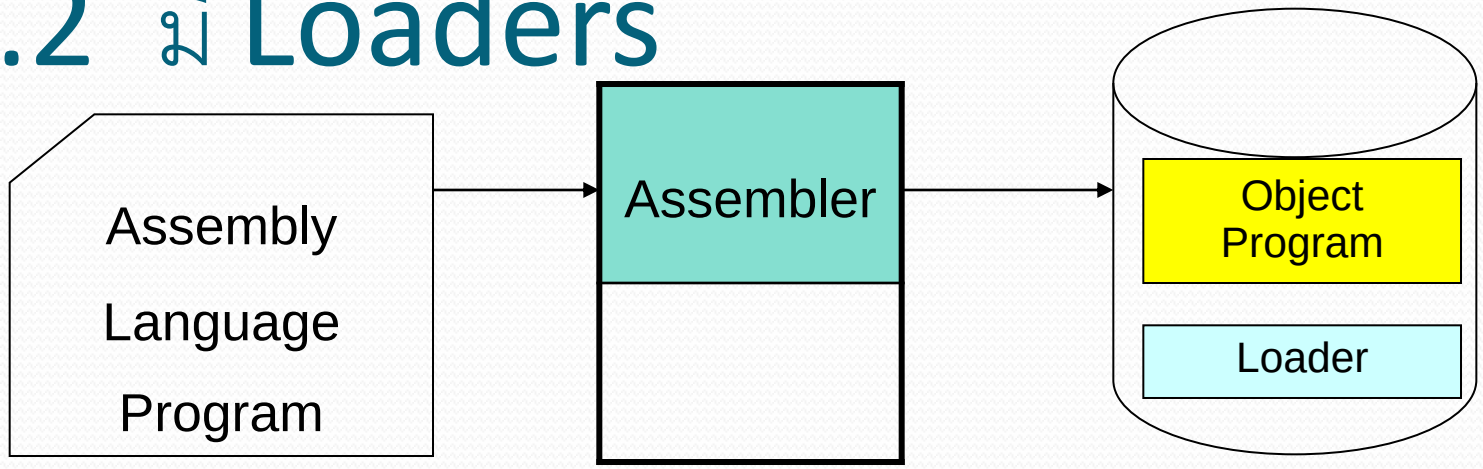


1.2.2 ไม่มี Loaders

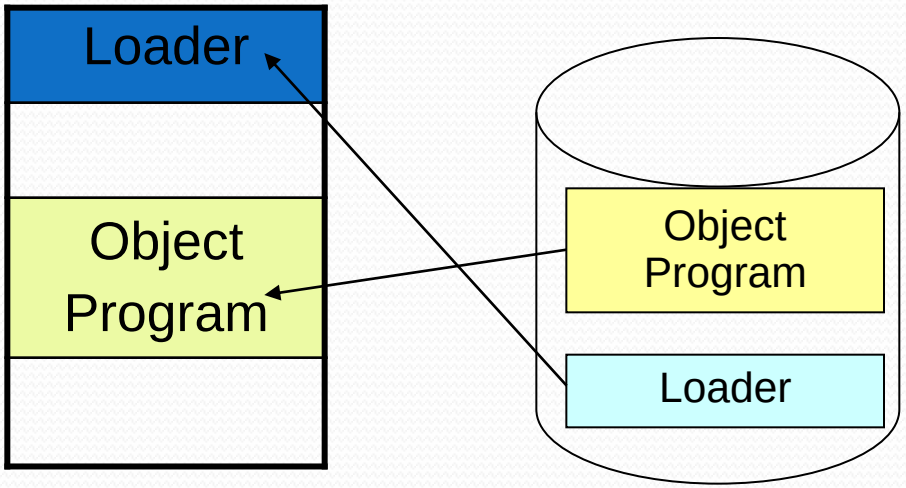


- เปลืองหน่วยความจำ
- ต้องทำการแปลงทุกครั้ง เมื่อต้องการ Execute

1.2.2 มี Loaders



- Loader เล็กกว่า ตัว Assembler ดังนั้นกิน หน่วยความจำน้อยกว่า สามารถใช้ Object Program ได้ใหญ่ขึ้น
- สามารถใช้โปรแกรมย่อยได้

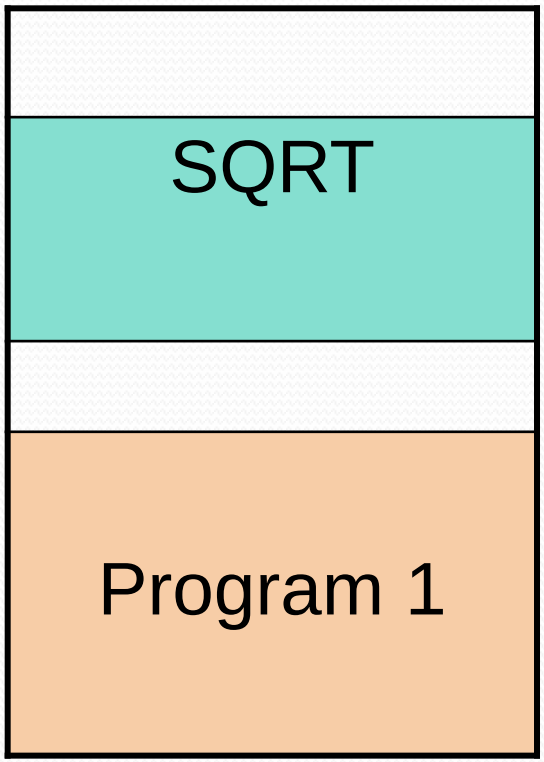


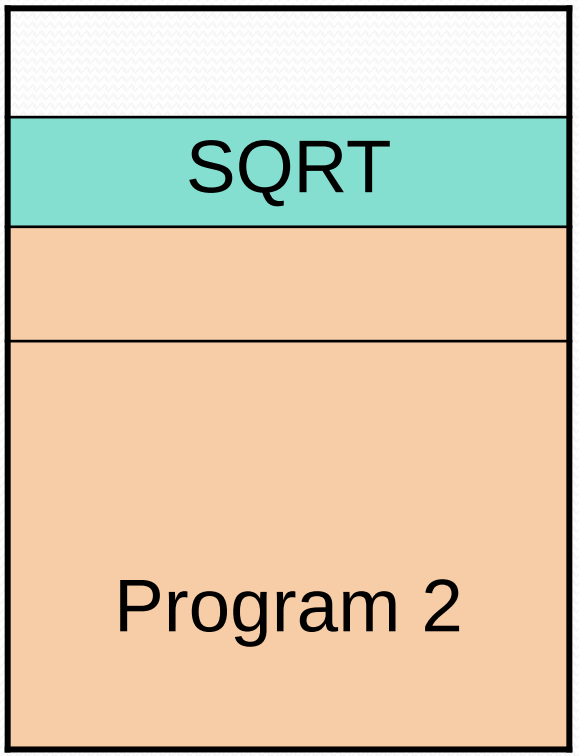
โปรแกรมย่อย (subroutines)

- Subroutine จะแบ่งเป็น 2 ประเภท คือ
  - Closed subroutines
    - จะถูกเก็บอยู่ภายนอก main program ซึ่งจะถูกเรียกเมื่อมีการเรียกใช้โดย main program ซึ่งจะทำการ link ไปเรียกใช้ subroutine อีกทีหนึ่ง ดังนั้นหน้าที่ของ main program จะมี 2 อย่างคือ Transfer control และ Transfer data
  - Open subroutines
    - หรือบางครั้งเรียกว่า Macro definition เป็นการ insert code ลงใน main program ดังนั้นหากมีการเรียกใช้ subroutine นั้น 4 ครั้ง ตัว code ของ subroutine ก็จะปรากฏใน 4 ที่ ภายใน program

ปัญหาของโปรแกรมย่อย

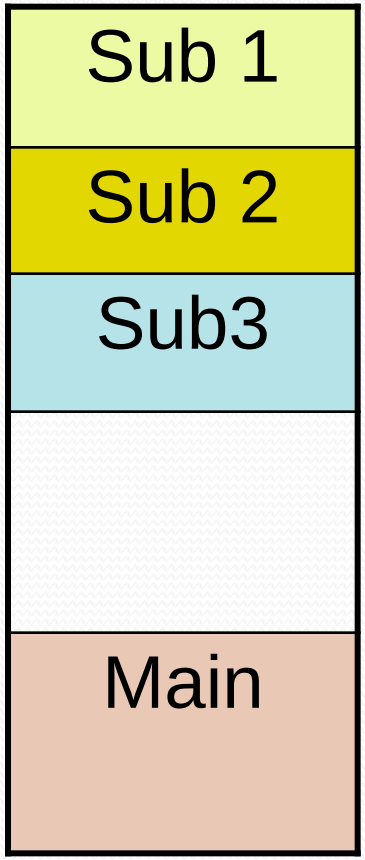
- โปรแกรม 1 (ขนาดเล็ก) มี Hole
- โปรแกรม 2 (ขนาดใหญ่) ทับโปรแกรมย่อย SQRT





หน้าที่ของ Loader

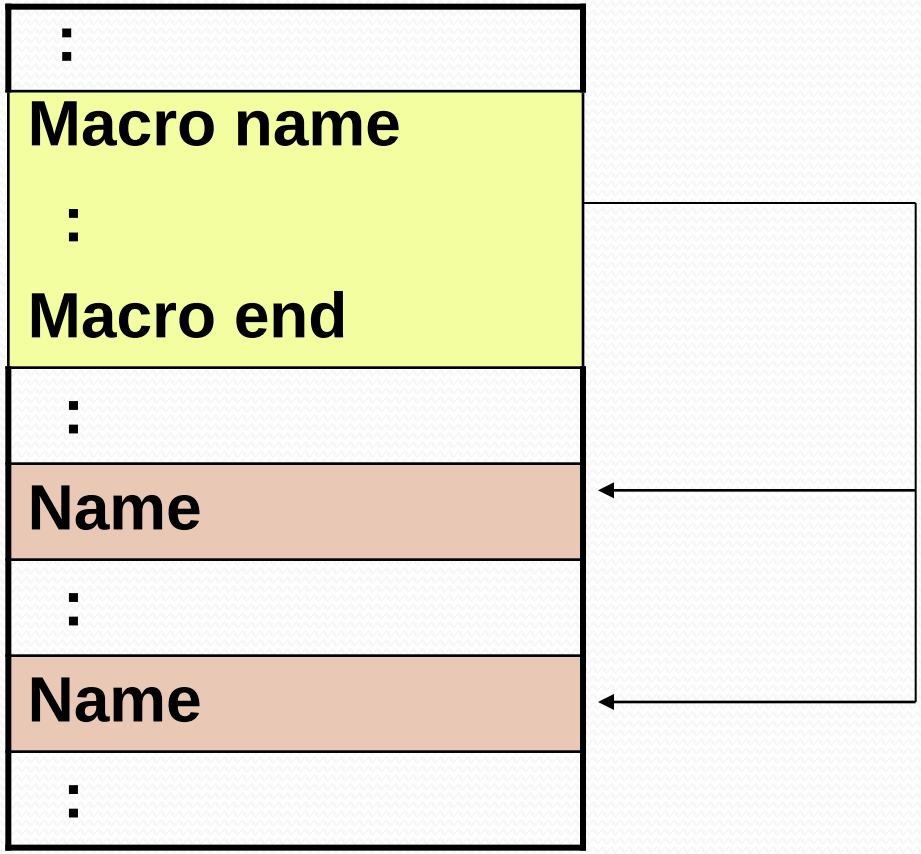
- จัดสรรพื้นที่ในหน่วยความจำให้กับโปรแกรม (Allocation)
- เชื่อมโยง Address ของ Symbolic ต่างๆ ระหว่าง Object (Linking)
- ปรับ Address ให้สอดคล้องกับหน่วยความจำ ในกรณีการอ้าง Address เป็นแบบ Relocate Address (Relocation)
- เรียกโปรแกรมจากดิสก์ มาไว้ยังหน่วยความจำ (Loading)



Time

- Execution Time ระยะเวลาในการ Execute โปรแกรมของผู้ใช้
- Assembly Time ระยะเวลาในการแปลงโปรแกรมภาษาแอสเซมบลี้ ให้เป็น Object Program
- Compile Time ระยะเวลาในการแปลงโปรแกรมภาษาชั้นสูง เช่น Pascal ให้เป็น Object Program
- Load Time ระยะเวลาในการโหลดโปรแกรมจากดิสก์ลงหน่วยความจำ และปรับแอดเดรส พร้อมให้โปรแกรมทำงาน

1.2.3 Macro Processor

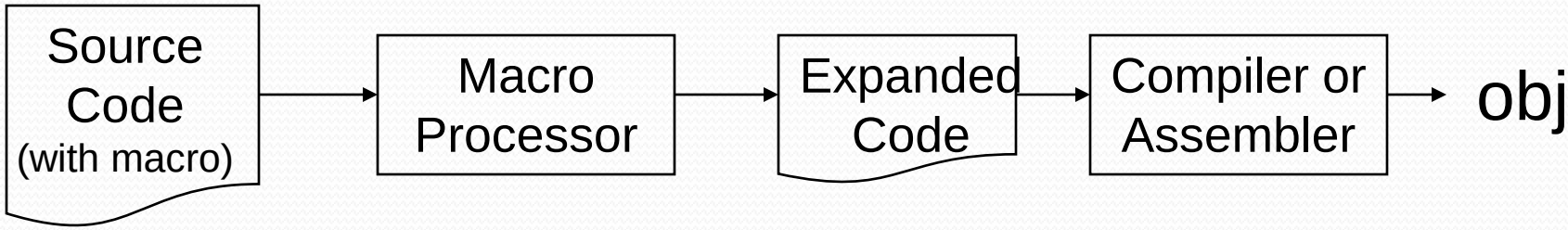


- แทนที่ ขยาย
- ทำโดย Macro Processor

ลดงานโปรแกรมเมอร์ ในการ Code โปรแกรม

Macro Processors

- Create macro definitions
- Save the macro definition
- Recognize macro calls
- Expand macro calls



Macro Processors

Source

```
STRG    MACRO
        STA     DATA1
        STB     DATA2
        STX     DATA3
        MEND
    .
STRG
    .
STRG
    .
    .
```

Expanded source

```
    .
    .
    .
        STA     DATA1
        STB     DATA2
        STX     DATA3
    .
        STA     DATA1
        STB     DATA2
        STX     DATA3
    .
```

Macro Processors

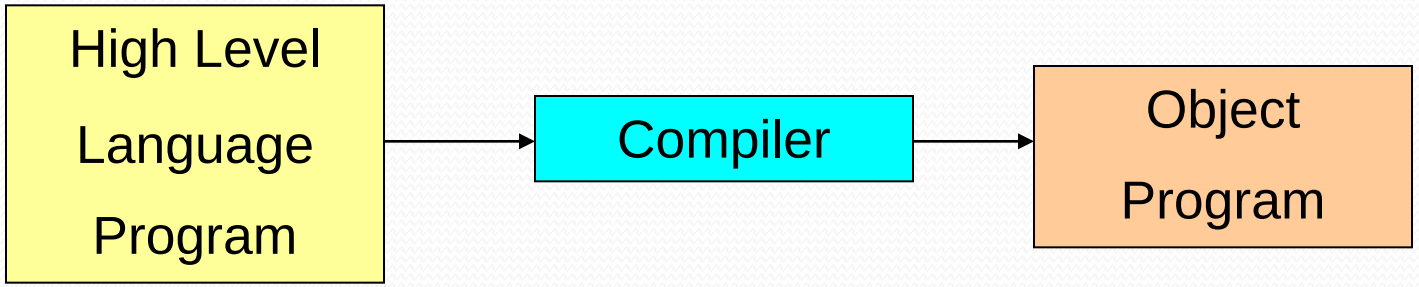
Source

```
STRG    MACRO  &a1, &a2, &a3
        STA    &a1
        STB    &a2
        STX    &a3
        MEND
  .
STRG    DATA1, DATA2, DATA3
  .
STRG    DATA4, DATA5, DATA6
  .
  .
```

Expanded souce

```
  .
  .
  .
    { STA    DATA1
    { STB    DATA2
    { STX    DATA3
  .
    { STA    DATA4
    { STB    DATA5
    { STX    DATA6
  .
```

1.2.4 Compiler



- แปล High level language ให้เป็น Object Program

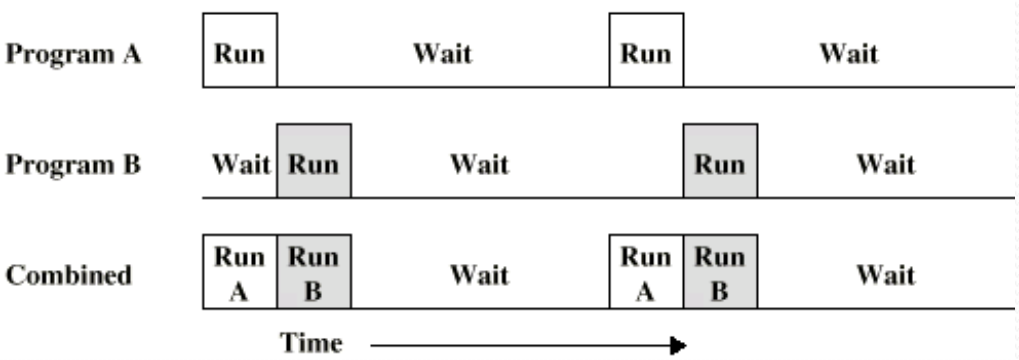
1.3 Operating System

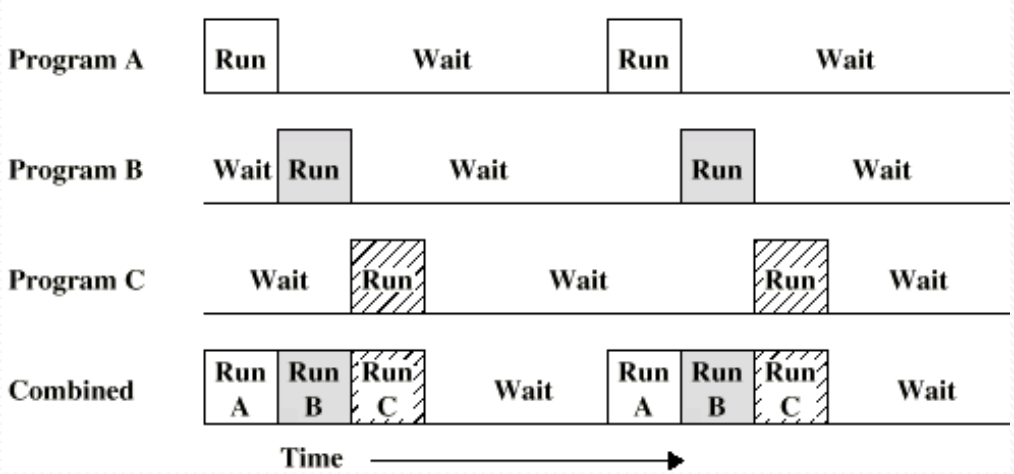
- ขั้นตอนในการ Execute โปรแกรมระบบเก่า เช่น ภาษา Fortran ยุคเริ่มแรก
  - ใส่โปรแกรมระบบ Compiler ของภาษา Fortran (กดปุ่ม)
  - ใส่ Source Program (กดปุ่ม)
  - หา Loader จาก Library ใส่ (กดปุ่ม)
  - หยิบ Object Cards มาใส่ (กดปุ่ม)
  - หา Subroutine Card มาใส่ (กดปุ่ม)
- ยุ่งยาก ใช้ลำบาก
- เสียเวลามากในการทำงาน 1 Job/Program

Batch Operating System

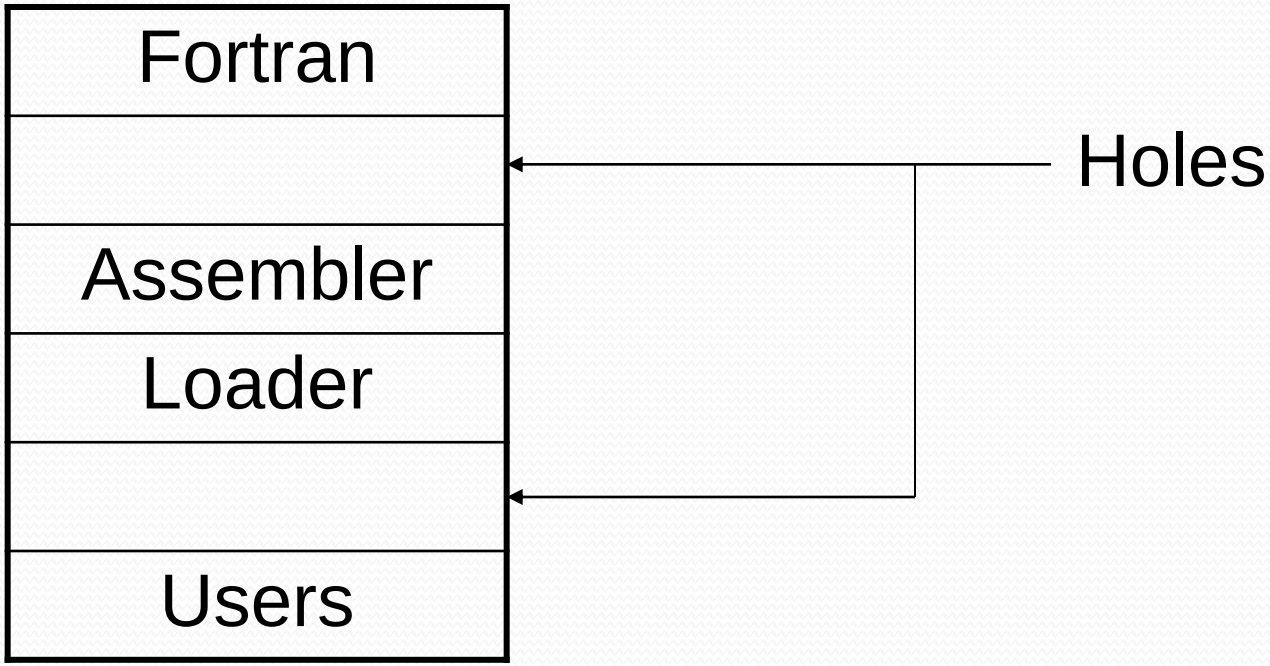
- โปรแกรมจำนวนมาก ถูก Loaded ลงหน่วยความจำ
- เป็นระบบ Multiprogramming (มีหลายโปรแกรมบรรจุภายในหน่วยความจำ CPU สามารถสวิทช์ไปทำโปรแกรมอื่นได้ ถ้าโปรแกรมที่กำลังทำงานอยู่ รอการใช้อุปกรณ์ I/O)
  - Multiprogramming with Fixed Tasks (MFT)
    (หน่วยความจำแต่ละโปรแกรมคงที่)
  - Multiprogramming with Variable Tasks (MFT)
    (หน่วยความจำแต่ละโปรแกรม แปรไปตามขนาดของโปรแกรม)

Multiprogramming





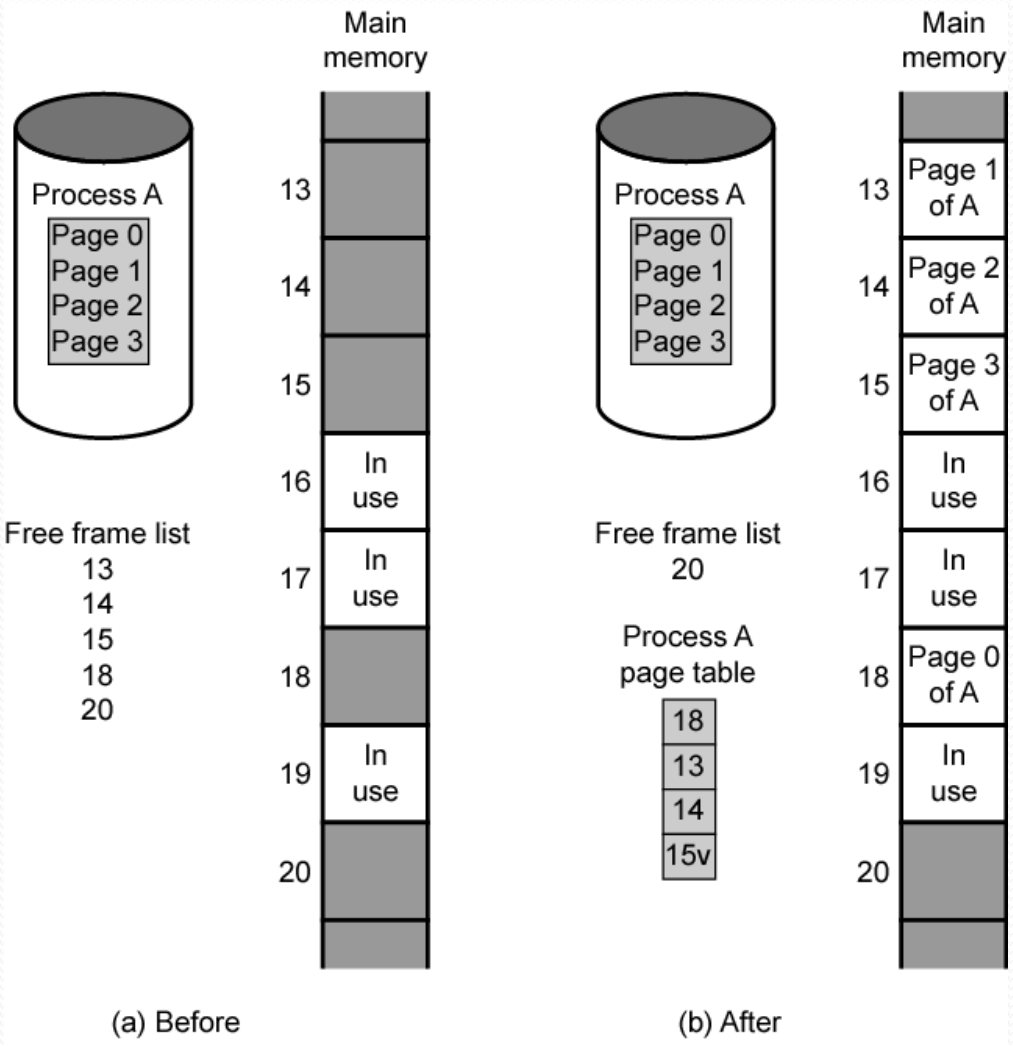
Fragmentation



การแก้ปัญหา Fragmentation

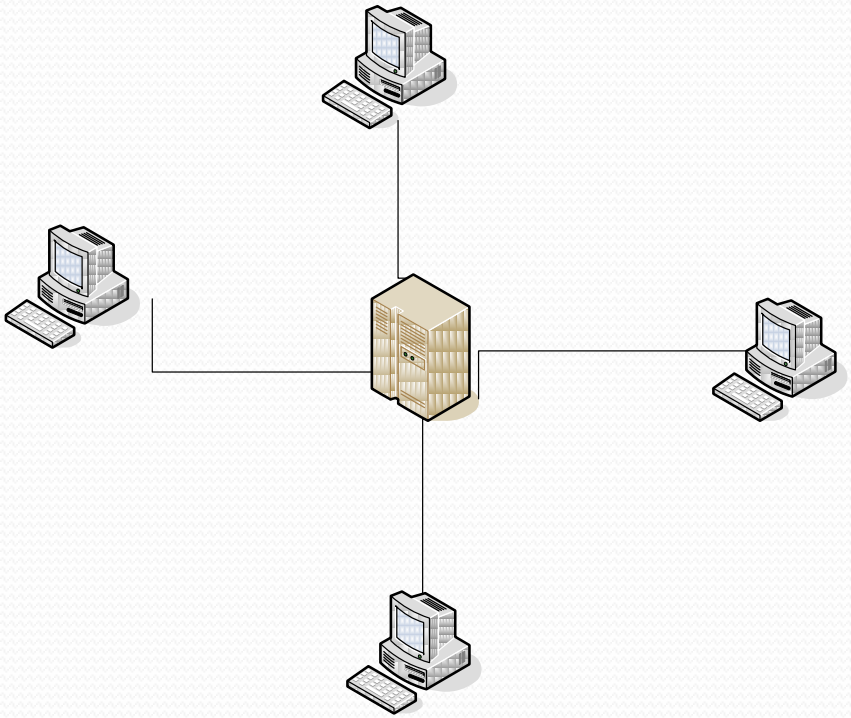
- Relocatable Partition
- Paging
- Segment

Paging



- ตัวอย่าง : สมมติ Process A มีพื้นที่ในหน่วยความจำ 4 Page OS ก็จะทำการหา Frame ที่ว่างมา 4 Frame โดยที่ไม่จำเป็นจะต้องอยู่ติดกันจาก Free Frame List จะมี Page Table สำหรับเก็บไว้ว่า Page ไหนถูกเก็บไว้ Frame ไหนของหน่วยความจำ

Time Sharing



แบ่งเวลาของ CPU ออกเป็นส่วน ให้กับ User แต่ละคน จนดูเหมือนว่า User แต่ละคนเป็นเจ้าของ CPU

หน้าที่หลักของ OS

- Job Sequencing, Job scheduling, traffic controller operation
- Input/Output programming
- Protection ifself from the user ; Protecting the user from other users
- Secondary storage management
- Error handling

Summary

- Loader มีหน้าที่ทำการ load object program (machine code) เพื่อเตรียมพร้อมเข้าสู่กระบวนการ execute การทำงานของ loader มีรูปแบบดังนี้เช่น
  - Load object program to main memory
  - Relocate address of object program
  - Link object programs
- Compiler มีหน้าที่แปลง source program ซึ่งเป็น High level language ให้เป็น object program

Summary

- Assembler คือ ตัวแปรภาษา assembly ที่ได้ output ออกมาเป็น Machine code หรือ Object program เพื่อเตรียมพร้อมให้ Loader นำไปทำการ Execute

- Macro Processor
  - Macro call เป็นการเรียกใช้ โดยอ้างถึงชื่อ Macro
  - Macro definition คือ ตัว code ที่ถูกเรียกใช้
  - Macro processor คือ system program ที่ทำหน้าที่ นำ Macro definition ไปวางแทนที่ ส่วนของ code ทีเรียกโดย Macro call

Summary

- Operating system ทำการจัดการเกี่ยวกับการจัดการทรัพยากร และบริการต่าง ๆ ของคอมพิวเตอร์ เช่น memory, processors, devices และ information OS เป็น system program มีหน้าที่จัดการทรัพยากรต่าง ๆ เช่น
  - Traffic controller
  - Scheduler
  - Memory management module
  - I/O programs
  - File system