

ตอนที่ 1

**ไมโครคอมพิวเตอร์
และ
Micro VAX II**

1. ไมโครคอมพิวเตอร์

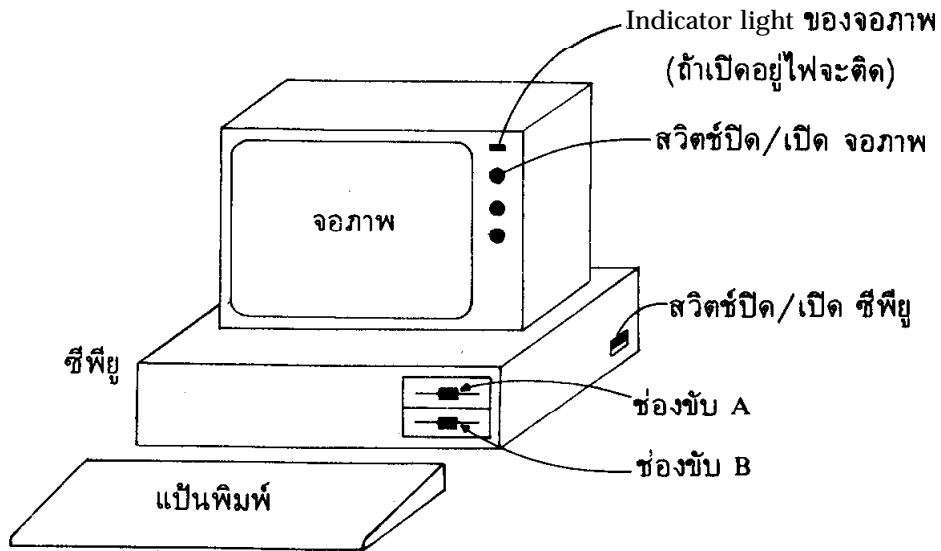
เครื่องไมโครคอมพิวเตอร์ที่ห้อง STB 104 เป็น IBM Compatible ประกอบด้วย

CPU 8088-2 มี RAM 640 Kb

ช่องขับดิสเกตต์ (disk drive) แบบ 2DD-360 Kb 2 ช่องขับ

จอโมโนโครม

แป้นพิมพ์มาตรฐาน QWERTY



แป้นพิมพ์ (Keyboard)

F1	F2
F3	F4
F5	F6
F7	F8
F9	F10

Function keys

~ `	! 1	@ 2	# 3	\$ 4	% 5	^ 6	& 7	* 8	(9) 0	- _	+ =	;/ '	- _
← ↑	Q	W	E	R	T	Y	U	I	O	P	{ [}]	Enter	
Ctrl	A	S	D	F	G	H	J	K	L	; :	" '	↵		
Shift	Z	X	C	V	B	N	M	< ,	> .	? /	Shift			
Alt	Space bar											Caps Lock		

Main Keyboard

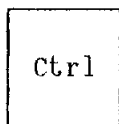
Visual indicator

Caps Lock	Num Lock	Scroll Lock
-----------	----------	-------------

Esc	Num Lock	Scroll Lock	Sys Req
7 Home	8 ↑	9 Pg Up	Prt Sc *
4 -	5	6 -	-
1 End	2 ↓	3 Pg Dn	+
0 Ins	.	Del	

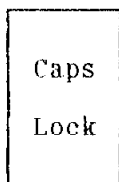
Numeric Keypad

1.2 คำอธิบายปุ่ม (Key) ในแป้นพิมพ์หลัก (Main Keyboard)



(Control)

ปุ่มนี้ใช้คู่กับปุ่มอื่น (โดยกดปุ่มนี้เข้าไว้ก่อนจึงกดปุ่มอื่นตาม แล้วปล่อย
พร้อมกันทั้งสองปุ่ม) เพื่อส่งรหัสควบคุมไปยังระบบ รหัสควบคุม
จะบอกระบบให้ปฏิบัติตามที่กำหนดไว้ก่อนแล้ว เช่นใน MS-DOS
เมื่อกด Ctrl-C จะมีผลทำให้เกิดการขัดจังหวะในการทำงานของ
Batch file



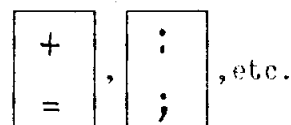
(Capitals lock)

เมื่อกดปุ่มนี้ไฟที่ CAPS LOCK จะติดหรือดับ ถ้าไฟติดแสดงว่าเมื่อกด
ปุ่มตัวอักษร จะได้อักษรตัวพิมพ์ใหญ่ (uppercase letter)
เช่น A, B, C, ... ถ้าไฟดับ เมื่อกดปุ่มตัวอักษร จะได้อักษรตัว
พิมพ์เล็ก (lowercase letter) เช่น a, b, c, ...

ในกรณีหลังนี้ถ้าต้องการอักษรตัวพิมพ์ใหญ่ ให้กดปุ่ม Shift เข้าไว้
แล้วกดปุ่มตัวอักษรที่ต้องการ



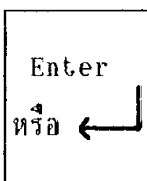
ปุ่มนี้ต้องใช้คู่กับปุ่มที่มี 2 ตัวอักษร เช่นปุ่ม




ถ้าต้องการอักขระตัวบน ต้องกดปุ่มนี้เข้าไว้ แล้วกดปุ่มที่ต้องการ
เช่นถ้าต้องการเครื่องหมายบวก (+) จะต้องกดปุ่ม Shift เข้าไว้
แล้วกดปุ่ม



แล้วปล่อยทั้งคู่



หรือ

ใช้เพื่อส่งคำสั่งระบบ (System command) ไปยังระบบ เพื่อให้
ระบบปฏิบัติตาม ทุกครั้งที่ใช้คำสั่งระบบ เมื่อพิมพ์คำสั่งเสร็จ
แล้วต้องกดปุ่มนี้ทุกครั้ง ในขณะที่ทำการ edit พิมพ์ จะกดปุ่มนี้
เมื่อต้องการขึ้นบรรทัดใหม่ ในหนังสือเล่มนี้จะแทนปุ่มนี้ด้วย 

Backspace
หรือ ←

เมื่อพิมพ์คำสั่งระบบ ใช้ปุ่มนี้เพื่อลบตัวอักษรที่อยู่หน้าเคอร์เซอร์ แต่ในขณะที่ทำการ edit ข้อมูลโดยใช้ WordStar ปุ่มนี้จะใช้เพื่อเลื่อนเคอร์เซอร์ไปทางซ้ายทีละตัวอักษร

Spacebar

คือแป้นขวางกลางสุดของแป้นพิมพ์ ไม่มีชื่อแสดงไว้ ใช้ในการตั้งตำแหน่งเคอร์เซอร์ตามต้องการ เมื่อทำการ edit ข้อมูลโดยใช้ WordStar ถ้า INSERT ON (พิมพ์แทรก) เมื่อกดปุ่มนี้ระหว่างข้อความจะทำให้เกิดช่องว่าง [blank(s)] ระหว่างข้อความ ถ้า INSERT OFF (พิมพ์ทับ) เมื่อกดปุ่มนี้ตัวอักษรที่เคอร์เซอร์อยู่จะถูกกลบไป

Alt

ปุ่มนี้ใช้คู่กับปุ่มอื่น เพื่อส่งคำสั่งไปยังระบบ เช่นในการบูท MS-DOS เมื่อเครื่องเปิดอยู่แล้ว เรากด Ctrl, Alt และ Del หรือในการส่งแฟ้มจากไมโครคอมพิวเตอร์ไปยัง

(Alternate)

Micro Vax II เรากด Alt, F10 และ S

Esc

กดปุ่มนี้ตามที่โปรแกรมจะระบุ

(Escape)

1.3 คำสั่งขยายปุ่มใน Numeric Keypad

ถ้ากดปุ่ม Num Lock (Number lock) ไว้ (ไฟที่ NUM LOCK จะติด) เมื่อกดปุ่มใด ๆ ใน Numeric Keypad จะได้อักษรตัวบนของปุ่ม

เมื่อทำการ edit ข้อมูลด้วย WordStar (กดปุ่ม Num Lock ให้อไฟดับ) ปุ่มต่าง ๆ จะมีความหมายตามที่ปรากฏด้านล่างของปุ่มดังต่อไปนี้

- ↑ เลื่อนเคอร์เซอร์ขึ้นไป 1 บรรทัด (ทำหน้าที่เหมือนกับ Ctrl-E)
- ↓ เลื่อนเคอร์เซอร์ลงมา 1 บรรทัด (ทำหน้าที่เหมือนกับ Ctrl-X)
- ← เลื่อนเคอร์เซอร์ไปทางซ้าย 1 ตัวอักษร (ทำหน้าที่เหมือนกับ Ctrl-S)
- เลื่อนเคอร์เซอร์ไปทางขวา 1 ตัวอักษร (ทำหน้าที่เหมือนกับ Ctrl-D)
- Pg Up (Page up) เพื่อดูข้อความในหน้าก่อนถึงหน้าที่ปรากฏอยู่บนจอ (ทำหน้าที่เหมือนกับ Ctrl-R)
- Pg Dn (Page down) เพื่อดูข้อความในหน้าถัดจากหน้าที่ปรากฏอยู่บนจอ (ทำหน้าที่เหมือนกับ Ctrl-C)
- Home ตั้งเคอร์เซอร์ให้อยู่ที่คอลัมน์ที่ 1 ของบรรทัดที่ 1
- End ตั้งเคอร์เซอร์ให้อยู่ที่คอลัมน์ที่ 1 ของบรรทัดที่ 14 (ทั้งหมดมี 15 บรรทัด)
- Ins (Insert) ทำให้ INSERT ON/OFF (ทำหน้าที่เหมือน Ctrl-V)
- Del (Delete) ใช้นับตัวอักษร 1 ตัวทางซ้ายของเคอร์เซอร์

2. Microsoft-Disk Operating System (MS-DOS)

2.1 ลักษณะทั่วไปของ MS-DOS

MS-DOS เป็นโปรแกรมควบคุมการปฏิบัติงานของไมโครคอมพิวเตอร์ของบริษัท IBM (IBM-PC) ทุกครั้งที่จะใช้เครื่องไมโครคอมพิวเตอร์ เครื่องจะต้องทำการบูท (boot) หรือบรรจุ (load) โปรแกรมควบคุมการปฏิบัติงานหรือโปรแกรมควบคุมระบบ (Operating System) ซึ่งเก็บอยู่ในแผ่นดิสเกตต์หรือใน Hard disk ลงในหน่วยความจำหลักของไมโครคอมพิวเตอร์ การบูทจะทำโดยอัตโนมัติเมื่อเปิดเครื่องคอมพิวเตอร์ ซึ่งในช่องทัพบมีแผ่นดิสเกตต์ที่มี MS-DOS อยู่

MS-DOS ประกอบด้วยโปรแกรม 3 ส่วน ซึ่งจะทำการควบคุมและจัดการทำงานทุกอย่างในระบบ โปรแกรมทั้ง 3 คือ

1. COMMAND.COM ทำหน้าที่เกี่ยวกับการรับคำสั่งจากผู้ใช้ผ่านทางแป้นพิมพ์ และทำการแปลความหมายของคำสั่งนั้น เพื่อทำการเรียกใช้ MSDOS.SYS อีกต่อหนึ่ง
2. MSDOS.SYS เป็นส่วนที่ใช้ในการเข้าถึง (access) รูทีนต่าง ๆ ของ MS-DOS

3. IO.SYS เป็นส่วนที่ทำหน้าที่ควบคุมอุปกรณ์รอบข้างของระบบ เช่น แป้นพิมพ์ จอภาพ เครื่องพิมพ์ เป็นต้น

2.2 การเริ่มต้น MS-DOS

2.2.1 การเริ่มต้น MS-DOS เมื่อเครื่องปิดอยู่

ให้ทำตามลำดับดังนี้

1. ใส่แผ่นดิสเกตต์ที่มี MS-DOS ในช่องขับ A แล้วปิดลิ้นคิให้เรียบร้อย
2. เปิดจอภาพ และเปิดสวิชเพื่อให้เครื่องเริ่มทำงาน จะเห็นไฟที่ช่องขับ A ติด

สักครู่จะปรากฏข้อความบนจอภาพให้ใส่วัน และเวลา (หลังจากเรากำหนดวันแล้ว)

2.2.1 การเริ่มต้น MS-DOS เมื่อได้เปิดเครื่องใช้งานอื่นมาก่อนแล้ว

เมื่อต้องการจะเปลี่ยนมาใช้ MS-DOS ให้ทำตามลำดับดังนี้

1. ใส่แผ่นดิสเกตต์ที่มี MS-DOS ในช่องขับ A แล้วปิดลิ้นคิให้เรียบร้อย
2. กดปุ่ม **Ctrl** และ **Alt** ซ้ำไว้ก่อน กด **Del** ตาม แล้วปล่อยทุกปุ่ม

การกระทำดังนี้เป็นาร reset เครื่องเหมือนกับการเปิดเครื่องใหม่โดยไม่ต้องปิดแล้ว เปิดสวิชใหม่ รอสักครู่จะปรากฏข้อความให้ใส่วันและเวลา

2.3 การกำหนดวันที่และเวลา

ข้อความบนจอภาพคือ

Current date is Tue 1-01-1980

Enter new date:, <---- เคอร์เซอร์อยู่ที่นี้

ให้พิมพ์วันตามรูปแบบดังนี้

MM-DD-YY

 (เดือน-วันที่-ปี ค.ศ.)

ตัวอย่าง ถ้าต้องการใส่วันที่ 5 มกราคม 1994 เราอาจพิมพ์อย่างใดอย่างหนึ่งได้ดังนี้

1-5-94

01-05-94

1/5/94

01/5/94

01/05/94

ถ้าใส่ไม่ถูกต้องตามรูปแบบจะปรากฏข้อความว่า invalid date แล้วเครื่องจะถามให้ใส่วันใหม่ เมื่อใส่วันถูกต้องจะปรากฏข้อความให้ใส่เวลา

Current time is 0:00:23.28

Enter new time:.

ให้พิมพ์เวลาตามรูปแบบดังนี้

ชั่วโมง:นาที:วินาที.หนึ่งในร้อยของวินาที

ตัวอย่าง ต้องการใส่เวลา 9.30 น. เราอาจพิมพ์

9:30:0.00

09:30:0.00

9:30:0

9:30

เมื่อใส่เวลาถูกต้องจะปรากฏข้อความบนจอบอกเกี่ยวกับ MS-DOS และจะปรากฏเครื่องหมาย A> (อ่านว่า A-prompt) แสดงว่าขณะนั้น MS-DOS กำลังติดต่อกับช่องรับ A และ MS-DOS พร้อมทั้งจะรับคำสั่งโดยให้ช่องรับ A เป็นเสมือนคลังคำสั่ง ถ้าเราใส่คำสั่งที่ไม่มีอยู่ในหน่วยความจำหรือไม่มีอยู่ในแผ่นดิสเกตต์ในช่องรับ A เครื่องจะบอกว่าไม่รู้จักคำสั่งนี้ด้วยข้อความ (ทั้งนี้อาจเป็นผลจากการกดแป้นพิมพ์ผิดตอนพิมพ์คำสั่งด้วย)

Bad command or filename

2.4 การกำหนดช่องขับที่จะใช้

ถ้าต้องการเปลี่ยนไปติดต่อกับแผ่นดิสเกตต์ในช่องขับ B เราทำได้โดย

คำสั่ง A>B:(E)

บนจอจะได้ B>

และถ้าต้องการกลับไปช่องขับ A ตามเดิมให้พิมพ์ A:(E) นั่นคือใช้คำสั่ง

คำสั่ง B>A:(E)

บนจอจะได้ A>

2.5 ชื่อแฟ้มในระบบ MS-DOS

ในการเรียกหรือกำหนดชื่อแฟ้ม จะมีส่วนประกอบทั้งหมด 3 ส่วนคือ

1. ช่องขับที่กำหนด (ช่องขับที่มีแผ่นดิสเกตต์ที่มีแฟ้มที่ระบุอยู่)
2. ชื่อแฟ้ม (filename)
3. ชนิดของแฟ้ม (file type) หรือส่วนขยายชื่อแฟ้ม (file extension)

รูปแบบ

d:filename.ext

โดยที่ d: เป็นช่องขับที่กำหนด เช่น A:, B: เป็นต้น

filename เป็นชื่อแฟ้มที่ประกอบด้วยตัวอักษร 1-8 ตัว ตัวอักษรที่ประกอบเป็นชื่อได้แก่ A-Z (ใช้ a-z ได้ด้วย คอมพิวเตอร์จะเปลี่ยนเป็นอักษรตัวใหญ่เอง) 0-9, #, @, :, %, ', (,), ^, และ &

.ext เป็นส่วนขยายประกอบด้วยตัวอักษร 1-3 ตัว จากตัวอักษรที่ใช้ตั้งชื่อแฟ้มได้ โดยปกติส่วนขยายชื่อแฟ้มนี้จะตั้งให้มีความหมายเฉพาะ เช่น

ส่วนขยาย	ความหมาย
.BAK	แฟ้มสำรองที่ถูกสร้างขึ้น (BAK ย่อมาจาก <u>back</u> up)
.BAT	แฟ้มของชุดคำสั่ง (BAT ย่อมาจาก <u>batch</u> file)
.COM	แฟ้มคำสั่งเครื่อง (COM ย่อมาจาก <u>command</u>)
.DAT	แฟ้มเก็บข้อมูลเข้า (DAT ย่อมาจาก <u>data</u>)
.EXE	แฟ้มภาษาเครื่องที่เครื่องจะปฏิบัติ (<u>execute</u>) ตาม ตอนวิ่งโปรแกรม
.FOR	แฟ้มภาษาฟอร์แทรน (FOR ย่อมาจาก <u>FOR</u> TRAN)
.LIB	คลังแฟ้ม (LIB ย่อมาจาก <u>library</u>)
.LST	แฟ้มเก็บ source program <u>listing</u> จาก MS-DOS
.LIS	แฟ้มเก็บ source program <u>listing</u> จาก Micro Vax II
.OBJ	แฟ้มเก็บ <u>object</u> program
.OUT	แฟ้มเก็บข้อมูลออก (OUT ย่อมาจาก <u>output</u> data)

ในการทำงานแต่ละชิ้นให้นักศึกษาใช้ชื่อแฟ้มดังนี้
filename ให้นักศึกษาตั้งเอง เช่น งานชิ้นที่ 1 อาจใช้ filename ว่า ONE
ส่วน .ext ให้ใช้ดังนี้

.FOR เช่น ONE.FOR สำหรับเก็บ source program ภาษาฟอร์แทรน

.DAT เช่น ONE.DAT สำหรับเก็บข้อมูลเข้า (input data)

.OUT เช่น ONE.OUT สำหรับเก็บข้อมูลออก (output data)

2.6 การใช้คำสั่ง DIR ในการดูแฟ้มที่เก็บไว้ในแผ่นดิสเกตต์

คำสั่ง A>DIR (E) (DIR ย่อมาจาก directory คือสารบัญแฟ้ม)
เครื่องจะแสดงรายละเอียดของแฟ้มต่าง ๆ ในแผ่นดิสเกตต์ที่อยู่ในช่องขับ A
รายละเอียดที่แสดงคือ ชื่อแฟ้ม ชนิดของแฟ้ม ขนาดของแฟ้ม (บอกเป็นจำนวนไบต์)
วันที่สร้างแฟ้ม และเวลาที่สร้างแฟ้ม

```

เช่น      COMMAND  COM      17664  3-08-83  12.00  p.
        ONE      FOR      650    6-12-66  11.30  p.
        ONE      DAT      :      :      :
        ONE      LST      :      :      :
        :      :      :

```

47 File(s) 87040 bytes free
 ↑ ↑
 จำนวนแฟ้มในแผ่นดิสเกตต์ จำนวนไบท์ที่เหลือในแผ่นดิสเกตต์

คำสั่ง A>DIR/W **(E)**
 จะแสดงเฉพาะรายชื่อแฟ้ม และเรียงชื่อแฟ้มในแนวขวาง บรรทัดละ 5 ชื่อ เช่น

```

COMMAND  COB      ONE  DAT      ONE FOR . . .
SYS      COM      ...

```

```

47 File(s)                  87040 bytes free

```

คำสั่ง A>DIR/P **(E)** เพื่อดูรายชื่อแฟ้มทีละหน้า (P = Page)

คำสั่ง A>DIR B: **(E)** เพื่อดูรายชื่อและรายละเอียดเกี่ยวกับแฟ้มใน
 แผ่นดิสเกตต์ในช่องขับ B

คำสั่ง A>DIR/W B: **(E)** เพื่อดูเฉพาะรายชื่อแฟ้มในแผ่นดิสเกตต์ในช่องขับ B

2.7 การค้นหาแฟ้มในแผ่นดิสเกตต์

เมื่อต้องการทราบว่าแฟ้มที่ต้องการอยู่ในแผ่นดิสเกตต์ในช่องขับที่ระบุหรือไม่ เช่น ต้องการทราบว่าแฟ้มชื่อ TEST.FOR อยู่ในแผ่นดิสเกตต์ในช่องขับ A หรือไม่ เราใช้คำสั่ง

คำสั่ง A>DIR TEST.FOR (E)

ถ้ามีแฟ้ม TEST.FOR ในแผ่นดิสเกตต์ในช่องขับ A เครื่องจะพิมพ์รายละเอียดของแฟ้ม (ชื่อแฟ้ม ชนิดของแฟ้ม ขนาดของแฟ้ม เป็นต้น) ออกมาให้ดู แต่ถ้าไม่พบจะมีข้อความบอกว่าไม่พบแฟ้มว่า

File not found

ถ้าต้องการดูแฟ้มที่มีชื่อ TEST ไม่ว่าจะป็นชนิดใด ใช้คำสั่ง

คำสั่ง A>DIR TEST (E)

เครื่องจะพิมพ์รายละเอียดของแฟ้ม TEST.FOR, TEST.DAT, TEST.OUT เป็นต้น

ถ้าต้องการดูแฟ้มที่มีชื่อขึ้นต้นด้วยอักษร B ทุกแฟ้ม ใช้คำสั่ง

คำสั่ง A>DIR B*.* (E)

อาจได้รายละเอียดของแฟ้มเช่น BACKUP.COM, BASIC.OUT เป็นต้น

ถ้าต้องการดูทุกแฟ้มที่มีชนิดเป็น .FOR ใช้คำสั่ง

คำสั่ง A>DIR *.FOR (E)

2.8 การขอลดสิ่งที่เก็บในแฟ้มโดยใช้คำสั่ง TYPE

คำสั่งภายในชื่อ TYPE จะพิมพ์สิ่งที่เก็บอยู่ในแฟ้มที่ระบุชื่อไว้ออกมาบนจอภาพ โดยเครื่องจะพิมพ์สิ่งที่เก็บไว้ไปเรื่อย ๆ จนจบแฟ้ม (ไม่ให้สิ่ง TYPE กับแฟ้มที่มีชนิดเป็น .OBJ และ .EXE ออกมาดูที่จอภาพ เพราะแฟ้มที่มี .ext เป็น .OBJ และ .EXE เป็น Binary file ไม่ใช่ ASCII file)

ในการหยุดจอภาพที่ใด ๆ ของแฟ้มใช้ Ctrl-S และกดปุ่มใด ๆ ก็ได้เพื่อให้จอภาพวิ่งต่อไป

ตัวอย่าง

คำสั่ง A>TYPE TEST.FOR (E)

ใช้เพื่อดูสิ่งที่เก็บในแฟ้ม TEST.FOR ในแผ่นดิสเกตต์ในช่องขับ A

ถ้าแผ่นดิสเกตต์ที่มีแฟ้ม TEST.FOR อยู่ในช่องขับ B เราใช้คำสั่ง

คำสั่ง A>TYPE B:TEST.FOR (E)

ถ้าต้องการพิมพ์สิ่งที่อยู่ในแฟ้มชนิด ASCII file ออกทางเครื่องพิมพ์ ใช้คำสั่ง

คำสั่ง A>TYPE B:TEST.FOR >PRN:(E) (PRN: คือเครื่องพิมพ์)

2.9 การเปลี่ยนชื่อแฟ้มโดยใช้คำสั่ง RENAME

คำสั่งภายในชื่อ RENAME จะเปลี่ยนชื่อแฟ้มที่มีอยู่ให้ชื่อใหม่ตามที่ระบุ

รูปแบบของคำสั่งคือ

RENAME ชื่อเก่า ชื่อใหม่

ตัวอย่าง

คำสั่ง A>RENAME TEST.FOR ONE-FOR (E)

ชื่อ TEST.FOR จะหายไปจาก directory และมีชื่อ ONE.FOR แทนจำนวนแฟ้มใน directory เท่าเดิม

2.10 การลบแฟ้มที่ไม่ต้องการออกจากแผ่นดิสเกตต์

คำสั่งภายในชื่อ ERASE หรือ DEL จะทำการลบแฟ้มที่ระบุไว้

ตัวอย่าง

ถ้าต้องการลบชื่อแฟ้มชื่อ TEST.FOR ในแผ่นดิสเกตต์ในช่องขับ A ใช้คำสั่ง

คำสั่ง A>DEL TEST.FOR (E)

หรือ A>ERASE TEST.FOR (E)

ถ้าต้องการลบชื่อแฟ้มชื่อ TEST.FOR ในแผ่นดิสเกตต์ในช่องขับ B ใช้คำสั่ง

คำสั่ง A>DEL B:TEST.FOR (E)

ถ้าต้องการลบชื่อแฟ้มชื่อ TEST ไม่ว่าจะป็นชนิดใดในแผ่นดิสเกตต์ในช่องขับ A ใช้คำสั่ง

คำสั่ง A>DEL TEST.* (E)

ถ้าต้องการลบแฟ้มชื่อใดก็ได้ที่มีชนิด .FOR ในแผ่นดิสเกตต์ในช่องขับ A ใช้คำสั่ง

คำสั่ง A>DEL *.FOR (E)

ถ้าต้องการลบทุกแฟ้มในแผ่นดิสเกตต์ในช่องขับ A ใช้คำสั่ง

คำสั่ง A>DEL *.* (E)

หมายเหตุ การลบแฟ้มออกนั้น จะต้องแน่ใจว่าไม่ต้องการใช้แฟ้มนั้นอีกแล้ว เพราะสิ่งที่ลบไปแล้วจะเรียกกลับมาไม่ได้ นอกจากพิมพ์เข้าไปเก็บใหม่ หรือเรียกจาก .BAK ของแฟ้มนั้น ๆ (ถ้ามี) มาเปลี่ยนแปลงแก้ไขได้

2.11 การทำสำเนาแฟ้มโดยใช้คำสั่ง COPY

คำสั่งภายในชื่อ COPY จะทำสำเนาแฟ้มที่มีอยู่เก็บเพื่อไว้ใช้ในแผ่นดิสเกตต์แผ่นเดียวกัน หรือทำสำเนาเพื่อเก็บในแผ่นดิสเกตต์แผ่นอื่นได้

ตัวอย่าง

ถ้าต้องการทำสำเนาแฟ้ม TEST.FOR เก็บในแผ่นเดิมในชื่อ NTEST.FOR ใช้คำสั่ง

คำสั่ง A>COPY TEST.FOR NTEST.FOR 0 E

ถ้าต้องการทำสำเนาแฟ้ม TEST.FOR เก็บในแผ่นดิสเกตต์ในช่องขับ B ในชื่อเดิมใช้คำสั่ง

คำสั่ง A>COPY TEST.FOR B: (E)

ถ้าต้องการทำสำเนาแฟ้ม TEST.FOR เก็บในแผ่นดิสเกตต์ในช่องขับ B ในชื่อ NTEST.FOR ใช้คำสั่ง

คำสั่ง A>COPY TEST.FOR B:NTEST.FOR (E)

ถ้าต้องการทำสำเนาแฟ้มที่มีชื่อขึ้นต้นว่า TIM จากแผ่นดิสเกตต์ในช่องขับ A มาเก็บในช่องขับ B ในชื่อเดิม ใช้คำสั่ง

คำสั่ง A>COPY TIM*. * B: (E)

ถ้าต้องการทำสำเนาแฟ้มทุกแฟ้มจากแผ่นดิสเกตต์ในช่องขับ A มาเก็บในแผ่นดิสเกตต์ในช่องขับ B ในชื่อเดิม ใช้คำสั่ง

คำสั่ง A>COPY *.* B: (E)

2.12 การฟอร์แมต (FORMAT) แผ่นดิสเกตต์ใหม่

แผ่นดิสเกตต์ที่จะนำมาใช้งานได้ เราต้องทำฟอร์แมต (FORMAT) แผ่นก่อน โดยใช้คำสั่งภายนอกชื่อ FORMAT

สิ่งที่ต้องระวังคือ การนำแผ่นดิสเกตต์ที่มีข้อมูลอยู่แล้วมาทำการฟอร์แมตใหม่ ข้อมูลเดิมจะหายทั้งหมด

ขั้นตอนการฟอร์แมตแผ่นดิสเกตต์ (โดยไม่บันทึก MS-DOS ลงในแผ่นดิสเกตต์ที่จะถูกฟอร์แมต)

1. ใส่แผ่น MS-DOS ในช่องขับ A แล้วทำการบูท MS-DOS จนได้ A> บนจอภาพ
2. ใส่แผ่นดิสเกตต์ที่ต้องการฟอร์แมตในช่องขับ B
3. ใช้ คำสั่ง A>FORMAT B: (E)

ถ้าต้องการบันทึก MS-DOS ลงในแผ่นดิสเกตต์ที่ทำการฟอร์แมตด้วย ให้ใช้คำสั่ง

คำสั่ง A>FORMAT B:/S (E)

ดังที่ได้กล่าวมาแล้วในหัวข้อ 2.1 ว่า MS-DOS ประกอบด้วยโปรแกรม 3 โปรแกรม คือ IO.SYS, MSDOS.SYS และ COMMAND.COM 2 โปรแกรมแรกเป็นแฟ้มที่ซ่อนอยู่ (hidden file) นั่นคือเมื่อใช้คำสั่ง DIR จะไม่เห็นรายชื่อใน directory ส่วนแฟ้มหลังจะเห็นรายชื่อใน directory

ให้ลองดู directory ในแผ่นดิสเกตต์ในช่องขับ B (แผ่นที่ทำฟอร์แมทแล้ว) ด้วยคำสั่ง A>DIR B: (E)

2.13 คำสั่งทั่วไปของ MS-DOS

คำสั่งของ MS-DOS ตั้งแต่ version 2 ขึ้นไป ประกอบด้วยคำสั่งภายใน (Internal:I) และคำสั่งภายนอก (External:E) รวมกันมากกว่า 40 คำสั่ง

คำสั่งภายใน คือคำสั่งที่เมื่อเราบูท MS-DOS คำสั่งเหล่านี้จะถูกเก็บอยู่ในหน่วยความจำหลัก เราใช้ได้ทันทีเมื่อต้องการโดยที่ไม่ต้องมีแผ่นดิสเกตต์ที่มี MS-DOS อยู่ เช่น คำสั่ง DIR, TYPE, DEL, ERASE, COPY เป็นต้น

ส่วน คำสั่งภายนอก คือ คำสั่งที่เราเก็บอยู่ในแผ่นดิสเกตต์ เมื่อต้องการใช้จะต้องเอาแผ่นดิสเกตต์ที่มี MS-DOS ใส่ในช่องขับ แล้วจึงจะเรียกใช้ เช่น คำสั่ง DISKCOPY, FORMAT เป็นต้น

ต่อไปนี้เป็นคำสั่ง ชนิดของคำสั่ง และหน้าที่ของคำสั่งอย่างย่อ ๆ (รายละเอียดอ่านได้จากหนังสือเล่มที่ 3 และเล่มที่ 4 ในบรรณานุกรม)

2.13.1 คำสั่งเกี่ยวกับแฟ้มและแผ่นดิสเกตต์

คำสั่ง	ชนิด	หน้าที่
ERASE	I	ลบแฟ้ม
RENAME	I	เปลี่ยนชื่อแฟ้ม
COPY	I	ทำสำเนาแฟ้ม
COMP	E	เปรียบเทียบแฟ้ม
DEL	I	ลบแฟ้ม
DIR	I	ดู directory
FORMAT	E	ทำการฟอร์แมตแผ่นดิสเกตต์เพื่อเตรียมไว้ใช้งาน
RECOVER	E	ซ่อมแฟ้มที่มี sector เสีย
DISKCOPY	E	ทำสำเนาแผ่นดิสเกตต์
DISKCOMP'	E	เปรียบเทียบแผ่นดิสเกตต์

2.13.2 คำสั่งเกี่ยวกับการพิมพ์

คำสั่ง	ชนิด	หน้าที่
TYPE	I	พิมพ์แฟ้มบนจอภาพตั้งแต่ต้นจนจบแฟ้ม
PRINT	E	ทำสำเนาแฟ้ม
MORE	E	หยุดถามหลังจากที่พิมพ์แฟ้มบนจอภาพทีละหน้า

2.13.3 คำสั่งตั้งระบบ

คำสั่ง	ชนิด	หน้าที่
DATE	I	กำหนดวัน
TIME	I	กำหนดเวลา
VERIFY	I	ตรวจสอบการเขียนข้อมูล
PROHPT	E	เปลี่ยนเครื่องหมายเตรียมพร้อม (prompt sign)

2.13.4 คำสั่งเกี่ยวกับ Subdirectory

คำสั่ง	ชนิด	หน้าที่
MKDIR	I	สร้าง directory
CHDIR	I	เปลี่ยน directory
RMDIR	I	ลบ directory
TREE	E	
PATH	I	

2.13.5 คำสั่ง Batch file

คำสั่ง	ชนิด
ECHO	I
PAUSE	I
IF	I
FOR	I
GO TO	I
REM	I
SHIFT	I

2.13.6 คำสั่งอื่น ๆ

คำสั่ง	ชนิด	หน้าที่
FIND	E	ค้นหาข้อความในแฟ้ม
SORT	E	เรียงลำดับข้อมูลในแฟ้ม
CLS	I	ลบจอภาพ จะได้ A> ที่บรรทัดบนสุดของจอ
GRAPHICS	E	อนุญาตให้พิมพ์รูปภาพพิกทางเครื่องพิมพ์
VER	I	ดูหมายเลข version ของ MS-DOS
VOL	I	ดู volume ของ MS-DOS

2.14 การสร้าง TEXT File บนจอภาพแล้วส่งไปเก็บในแผ่นดิสเกตต์

คำสั่ง A>COPY CON: SAMPLE.TXT (E)

พิมพ์สิ่งที่ต้องการเก็บในแฟ้ม SAMPLE.TXT

Ctrl-Z (E)

SAMPLE.TXT คือชื่อแฟ้มที่ต้องการใช้เก็บข้อความที่ต้องการ แฟ้มนี้จะเก็บอยู่ในแผ่นดิสเกตต์ในช่องขับ A

บนจอภาพจะมีข้อความว่า "1 File(s) copied"

ให้ลองเรียกสิ่งที่เก็บในแฟ้ม SAMPLE.TXT ออกมาด้วยคำสั่ง

คำสั่ง A>TYPE SAMPLE.TXT (E)

2.15 การใช้คำสั่ง COPY ในการรวมแฟ้มหลายแฟ้มเข้าด้วยกัน

คำสั่ง A>COPY TEST1.TXT+TEST2.TXT TEST3.TXT (E)

ผลของคำสั่งคือ แฟ้ม TEST3.TXT มีข้อความของแฟ้ม TEST1.TXT และของ TEST2.TXT รวมกัน ทั้งนี้อาจรวมแฟ้มมากกว่า 2 แฟ้มก็ได้

คำสั่ง A>COPY TEST1.TXT+TEST2.TXT+TEST3.TXT (E)

ผลของคำสั่งคือ แฟ้ม TEST1.TXT มีข้อความของทั้ง 3 แฟ้มรวมกัน

3. WordStar (WS)

WordStar เป็นโปรแกรมสำเร็จรูปสำหรับใช้งานด้านประมวลผลคำ (Word processing) นั่นคือ WordStar เป็น TEXT editor อย่างหนึ่ง WordStar ทำงานภายใต้โปรแกรมควบคุมการปฏิบัติงานที่มีชื่อว่า CP/M, MS-DOS

3.1 ขั้นตอนการปฏิบัติงาน

ขั้นที่ 1 ใส่แผ่น WS เข้าในช่องขับ A เปิดซีพียู เปิดจอภาพ เครื่องจะทำการบูท MS-DOS เข้าไปเก็บในหน่วยความจำหลักโดยอัตโนมัติ

ขั้นที่ 2 เครื่องจะแสดงข้อความบนจอ พร้อมทั้งถามวันเวลา ให้ใส่วันเวลาตามจริง (ดูวิธีการในหัวข้อ 2.3) ถ้าทำถูกต้องจะได้เครื่องหมาย A) ซึ่งแสดงว่าเครื่องพร้อมจะรับคำสั่งเพื่อสั่ง MS-DOS แล้ว

ขั้นที่ 3 พิมพ์ WS (E)

เครื่องจะเรียกโปรแกรม WordStar ออกมาให้เราใช้งานตามต้องการ โปรแกรม WordStar จะมีเมนู (Menu) ต่าง ๆ ซึ่งในแต่ละเมนูจะมีคำอธิบายความหมายของตัวเลือกที่จะเลือกใช้ เพื่อให้โปรแกรมทำตามที่ต้องการได้

ในขั้นนี้จะอธิบายเพียงบางเมนูที่จำเป็นต้องใช้ในการสร้างแฟ้ม การแก้ไขแฟ้ม และการเก็บแฟ้มในแผ่นดิสเกตต์

3.2 เมนูต่าง ๆ ที่จะใช้

3.2.1 Opening Menu

Opening Menu คือ เมนูแรกหลังจากที่พิมพ์ PS **(E)** ซึ่งมีคำอธิบายความหมายของคำสั่งต่าง ๆ ที่เราอาจเลือกใช้ได้จากเมนูดังนี้

```
not editing
<< ( O P E N I N G M E N U ) >>
---Preliminary Commands--- | --File Commands-- | -System Commands-
L Change logged disk drive  | | R Run a program
F File directory now ON    | P PRINT a file      | X EXIT to system
H Set help level           | | -WordStar Options-
---Commands to open a file--- | E RENAME a file | T Run TelMerge
D Open a document file     | O COPY a file      | M Run MailMerge
N Open a non-document file | Y DELETE a file   | S Run SpellStar
```

เมื่อต้องการใช้งานใด ให้กดตัวอักษร 1 ตัวตามเมนู (ไม่ต้องกดปุ่ม Enter) ตัวอักษรแต่ละตัวจะมีความหมายดังนี้

- L เพื่อบอกให้เครื่องทราบว่าต้องการให้ช่องขับใดเป็นช่องขับที่กำลังใช้งานอยู่ เช่น ช่องขับ A หรือ ช่องขับ B เป็นต้น
- F สำหรับให้เครื่องแสดงหรือไม่แสดงรายชื่อแฟ้มที่มีอยู่ในแผ่นดิสเกตต์ในช่องขับที่กำลังใช้งานให้เห็นบนจอภาพ
- N เป็นคำสั่งให้เปิดแฟ้มชนิด non-document (ให้นักศึกษาใช้ตัวนี้ตลอดการสร้าง file.FOR และ file.DAT) ซึ่งคือแฟ้มที่เก็บสิ่งที่ต้องการใช้งานร่วมกับโปรแกรมอื่นทางคอมพิวเตอร์ เช่น ข้อมูลเข้า โปรแกรมคอมพิวเตอร์ (แฟ้มนี้จะเหมือนกับถูกสร้างด้วย EDLIN ใน MS-DOS)

- D เป็นคำสั่งให้เปิดแฟ้มชนิด document ใช้ในการพิมพ์เอกสาร เช่น จดหมาย ตำรา หนังสือ ต้องการการจัดการแฟ้มให้สวยงาม ชิดขอบขวา เป็นต้น
- P เป็นคำสั่งให้พิมพ์แฟ้มที่ต้องการออกทางเครื่องพิมพ์
- E เป็นคำสั่งให้เปลี่ยนชื่อแฟ้มจากชื่อเดิมเป็นชื่อใหม่
- O เป็นคำสั่งให้ทำสำเนาแฟ้ม (Copy file) เพื่อสำรองเก็บเอาไว้
- Y เป็นคำสั่งให้ลบแฟ้มออกจากแผ่นดิสเกตต์ โดยแฟ้มที่ลบจะต้องไม่เป็นแฟ้มที่กำลังใช้งานอยู่
- R เป็นคำสั่งให้วิ่งโปรแกรมต่าง ๆ ภายใต้ WordStar ได้ เช่น STAT, MBASIC
- X เป็นคำสั่งให้ออกไปสู่ระบบ คือไปที่ A>

ตัวอย่าง เมื่อต้องการสร้าง file.FOR

กด N (เพื่อสร้างแฟ้มชนิด Non-document ใหม่ หรือเรียกแฟ้มเก่าที่มีอยู่แล้ว ออกมาทำงาน

โปรแกรม WordStar จะถามชื่อแฟ้ม เราพิมพ์ชื่อแฟ้มที่ต้องการ เช่นต้องการแฟ้ม TEST.FOR

ENTER FILE NAME:TEST.FOR (E)

จะได้ Main menu (ถ้ากด (E) อย่างเดียวโดยไม่ระบุชื่อแฟ้ม โปรแกรมจะส่งกลับไปยัง Opening menu ใหม่)

3.2.2 Main Menu

เมนูนี้ใช้เมื่อจะพิมพ์ข้อความต่าง ๆ คำสั่งในเมนูนี้จะใช้ในการเลื่อนเคอร์เซอร์ และข้อความในแบบต่าง ๆ เพื่อช่วยในการพิมพ์

```

A:OR223.POR FC=1 FL=I COL 01          INSERT ON
<<<   M A I N   M E N U   >>>
--Cursor Movement--  | -Delete- | -Miscellaneous- | -Other Menus-
^S char left ^D char right | ^G char | ^I lab ^B Reform | (from Main only)
^A word left ^F word right | DEL chr lf | ^V INSERT ON/O,F,F | ^J Help ^K Block
^E line up ^X line down | ^T word rt | ^L Find/Replce again | ^Q Quick ^P Print
--Scrolling--        | ^Y line | RETURN End paragraph | ^O Onscreen
^Z line down ^W line up |          | ^N Insert a RETURN |
^C screen up ^R screen down; | ^U Stop a command |

```

ถ้าพิมพ์ที่ระบุเป็นแฟ้มเก่า ข้อความในแฟ้มจะปรากฏทีละ 15 บรรทัดถัดจากเมนูนี้
ความหมายของตัวเลือกต่าง ๆ คือ

- ^S (กดปุ่ม Ctrl แล้วยี้ แล้วกดตัว S ตาม)
ใช้เลื่อนเคอร์เซอร์ไปทางซ้าย 1 ตำแหน่ง
- ^D ใช้เลื่อนเคอร์เซอร์ไปทางขวา 1 ตำแหน่ง
- ^A ใช้เลื่อนเคอร์เซอร์ไปทางซ้าย 1 คำ
- ^F ใช้เลื่อนเคอร์เซอร์ไปทางขวา 1 คำ
- ^E ใช้เลื่อนเคอร์เซอร์ขึ้นไปยังบน 1 บรรทัด
- ^X ใช้เลื่อนเคอร์เซอร์ลงมาข้างล่าง 1 บรรทัด
- ^W ใช้เลื่อนข้อความลง 1 บรรทัด
- ^Z ใช้เลื่อนข้อความขึ้น 1 บรรทัด
- ^R ใช้เลื่อนข้อความขึ้น 1 หน้ากระดาษ
- ^C ใช้เลื่อนข้อความลง 1 หน้ากระดาษ

- ^G ใช้ลบทีละ 1 ตัวอักษร ณ ตำแหน่งที่เคอร์เซอร์อยู่
- DEL ใช้ลบทีละ 1 ตัวอักษรทางซ้ายของเคอร์เซอร์
- ^T ใช้ลบทีละคำ
- ^Y ใช้ลบบรรทัด ณ บรรทัดที่เคอร์เซอร์อยู่
- ^V เป็นคำสั่งให้พิมพ์แทรก (INSERT ON) และยกเลิกการพิมพ์แทรก
(คำ INSERT ON จะหายไปจากบรรทัดแรกของจอภาพ)
- ^N ใช้เปิดบรรทัดเหนือบรรทัดที่เคอร์เซอร์อยู่
การเปิดบรรทัดอาจทำได้โดยการตั้งเคอร์เซอร์หลังข้อความในบรรทัดบน แล้วกดปุ่ม Enter จะได้บรรทัดว่างก่อนบรรทัดถัดไป เราสามารถแทรกข้อความได้
- ^U ให้หยุดคำสั่งต่าง ๆ ก่อนจะทำงาน

เมื่อได้บันทึกสิ่งที่ต้องการลงในแฟ้มเรียบร้อยแล้ว เราจะจัดเก็บแฟ้มนั้นลงในแผ่นดิสเกตต์ได้โดยการไปที่ Block Menu

กด ^K (Ctrl-K) โดยไม่ต้องกด Enter

โปรแกรมจะไปที่ Block Menu

3.2.3 Block Menu

เมนูนี้เป็นเมนูสำหรับการจัด block ในข้อความต่าง ๆ เพื่อเคลื่อนย้ายไปยังส่วนต่าง ๆ หรือเพื่อจัดการอย่างอื่น ๆ ตามต้องการในการบันทึกและพิมพ์เอกสาร

```

^K      A:OR223.FOR FC=5 FL=3 COL 01          INSERT ON
          <<<  BLOCK MENU  >>>
-Saving Files- -Block Operations- -File Operations- | -Other Menus-
S Save & resume | B Begin K End | R Read P Print | (from Main only)
D Save--done | H Hide / Display | O Copy E Rename | ^J Help ^K Block
X Save & exit | C Copy Y Delete; J Delete | ^Q Quick ^P Print
Q Abandon file | V Hove N Write | -Disk Operations | ^O Onscreen
-Place Markers- ; N Column now OFF | L Change logged disk | Space Bar returns
0-9 set/hide 0-9 | | F Directory now OFF | you to Main Menu.

```

ความหมายของตัวเลือกในเมนูที่มีดังต่อไปนี้คือ

- S(^KS) ให้บันทึกแฟ้มลงในแผ่นดิสเกตต์ แล้วยังคงใช้แฟ้มเดิมต่อ
- D(^KD) ให้บันทึกแฟ้มลงในแผ่นดิสเกตต์ แล้วเลิกทำงานในแฟ้มนั้น ๆ แต่ยังคงอยู่ใน Opening Menu
- X(^KX) ให้บันทึกแฟ้มลงในแผ่นดิสเกตต์ แล้วออกจาก WordStar ไปยังระบบ ได้ A>
- Q(^KQ) ให้เลิกทำงานในแฟ้มนั้น โดยไม่มีการบันทึกลงในแผ่นดิสเกตต์
- P(^KP) ให้พิมพ์ (print) แฟ้มออกทางเครื่องพิมพ์ โดยต้องตอบคำถามต่าง ๆ เพื่อให้พิมพ์ตามที่ต้องการ
- O(^KO) ให้ทำสำเนา (copy) แฟ้มที่ต้องการไว้ในแผ่นดิสเกตต์
- E(^KE) ให้เปลี่ยนชื่อ (rename) แฟ้มจากชื่อเดิมไปเป็นชื่อใหม่
- J(^KJ) ให้ลบ (delete) แฟ้มออกจากแผ่นดิสเกตต์
- L(^KL) ให้ทำการกำหนดว่าช่องขับไดคจะเป็นช่องขับที่กำลังใช้งาน (logged drive)
- F(^KF) ใช้ในการเปิด/เปิด directory ซึ่งแสดงรายชื่อแฟ้มในช่องขับที่กำลังใช้งาน
- B(^KB) เพื่อกำหนดจุดเริ่มต้น block
- K(^KK) เพื่อกำหนดจุดสิ้นสุด block และใช้ลบ ^KB ซึ่งก่อดินที่
- C(^KC) เพื่อทำสำเนา block
- V(^KV) เพื่อเคลื่อนย้าย block
- Y(^KY) เพื่อลบ block
- W(^KW) เพื่อพิมพ์ block ลงไปเก็บในแฟ้มที่ระบุ
- R(^KR) เพื่ออ่านแฟ้มจากแผ่นดิสเกตต์เข้ามาเก็บในแฟ้มที่กำลังทำงานอยู่

3.3 Flag Character

Flag character คือตัวอักขระพิเศษที่อยู่ทางขวามือของจอภาพ เพื่อแสดงสถานะของข้อความที่พิมพ์ในแต่ละบรรทัดให้เราทราบ เช่น

เครื่องหมาย < แสดงว่า เราได้กดปุ่ม Enter เมื่อจบบรรทัดนั้น ๆ แล้วบรรทัดถัดไปถือเป็นบรรทัดใหม่

เครื่องหมาย + แสดงว่า บรรทัดนั้นยาวเลยออกไปจาก Screen แรกบนจอภาพ

เครื่องหมาย : แสดงว่า บรรทัดบนจอภาพนั้นอยู่ก่อนบรรทัดที่มีข้อความ

เครื่องหมาย . แสดงว่า บรรทัดบนจอภาพนั้นอยู่หลังบรรทัดที่มีข้อความ

4. ขั้นตอนการคอมไพล์โปรแกรมโดยใช้ Microsoft FORTRAN Compiler

v3.31

4.1 เมื่อ Microsoft FORTRAN Compiler V3.31 อยู่ในแผ่นดิสเกตต์ 2 แผ่น

คอมไพเลอร์แผ่นที่ 1 มี FOR1 และ PAS2

คอมไพเลอร์แผ่นที่ 2 มี LINK และ LIB (libraries)

ขั้นเตรียมการ

ช่องขับ A ใส่คอมไพเลอร์แผ่นที่ 1

ช่องขับ B ใส่แผ่นดิสเกตต์ที่มี Source program (file.FOR ที่จะคอมไพล์)
และมีแฟ้มข้อมูลเข้า (file.DAT)

ลองตรวจดู directory ในแผ่นดิสเกตต์ในช่องขับ A จะเห็นว่า มี FOR1 และ PAS2 อยู่ โดยใช้คำสั่งต่อไปนี้

คำสั่ง A>DIR (E)

คำสั่ง A>DIR B:(E)

คำสั่ง A>TYPE B:TEST.FOR (E)

คำสั่ง A>TYPE B:TEST.DAT (E)

เพื่อตรวจดูว่าโปรแกรมในแฟ้ม

TEST.FOR และข้อมูลเข้าในแฟ้ม

TEST.DAT พร้อมทั้งจะถูกคอมไพล์และ

วิ่งโปรแกรม

คำสั่ง A>B:(E) เพื่อเปลี่ยนจากช่องขับ A เป็นช่องขับ B จะได้ B>

ขั้นที่ 1

เริ่มทำขั้นที่ 1 ของการคอมไพล์โดยเรียกโปรแกรม FOR1 จากแผ่นดิสเกตต์ใน
ช่องขับ A ด้วยคำสั่ง

คำสั่ง B>A:FOR1 (E)

คอมไพเลอร์จะถามคำถาม 4 คำถาม โดยถามทีละคำถาม (ตัวอย่างคำตอบคือข้อความที่ขีดเส้นใต้)

Source filename [L.FOR]: TEST (E) ---> คอมไพเลอร์จะหาโปรแกรมใน
แฟ้ม TEST.FOR ในช่องขับ B
Object filename [TEST.OBJ]: (E) ---> สอมรับที่ a TEST.OBJ และเก็บ
ในช่องขับ B
Source listing [NUL.LST]: TEST (E) ---> เพื่อให้คอมไพเลอร์สร้าง
แฟ้ม TEST.LST เก็บในแผ่นดิสเกตต์ในช่องขับ B
Object listing [NUL.COD]: (E) ---> ไม่เก็บ Object listing

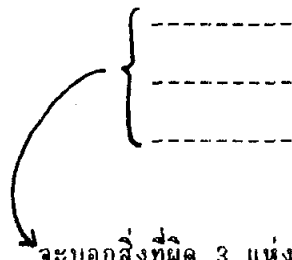
คอมไพเลอร์จะทำการคอมไพล์โปรแกรมในแฟ้ม TEST.FOR ซึ่งอยู่ในแผ่นดิสเกตต์
ในช่องขับ B

ถ้าโปรแกรมไม่มี Compile-time error บนจอภาพจะปรากฏข้อความว่า

Pass one No Errors Detected.

ถ้าจอภาพปรากฏข้อความ เช่น

Pass one 3 Errors Detected.



จะบอกสิ่งที่มี 3 แห่งที่คอมไพเลอร์ตรวจพบ นักศึกษาจะดูสิ่งที่มีขีดในคำสั่งที่ขีดได้
โดยดูในแฟ้ม TEST.LST ซึ่งถูกบันทึกในแผ่นดิสเกตต์ในช่องขับ B โดยใช้คำสั่งต่อไปนี้

คำสั่ง B>TYPE TEST.LST (E)

(ให้ Ctrl-S หยุดจอภาพ และกดปุ่มใด ๆ เพื่อพิมพ์ต่อ)

ให้นักศึกษาแก้ไขโปรแกรมในแฟ้ม TEST.FOR โดยใช้ Word Star เมื่อแก้ไขเสร็จแล้วกลับไปทำขั้นที่ 1 ซ้ำ จนได้ข้อความบนจอว่า Pass one No Errors Detected. แล้วทำขั้นที่ 2 ต่อ

วิธีแก้ไขโปรแกรมในแฟ้ม TEST.FOR โดยใช้ Word Star

คำสั่ง B>WS (E)

จะได้ Opening menu

ให้กด N และเมื่อถูกถามชื่อแฟ้มที่ต้องการแก้ไข ในที่นี้พิมพ์ TEST.FOR (E)

จะได้ Main menu พร้อมกับส่วนแรกของแฟ้ม TEST.FOR

เมื่อแก้ไขผิดครบแล้ว ให้ทำการเก็บแฟ้มนั้น โดยกด Ctrl-K แล้วตามด้วย X แล้วกลับไปทำขั้นที่ 1 ใหม่ ทำการคอมไพล์จนกว่าจะไม่มี compile-time error คือได้ข้อความบนจอว่า

Pass one No Error Detected.

25 Source lines.

↙ (บอกจำนวนบรรทัดในแฟ้มนั้น ๆ)

หมายเหตุ ทุกครั้งที่ทำการเก็บแฟ้ม เครื่องจะบันทึกแฟ้มที่ถูกแก้ไขด้วยตนเองเข้าข้อความในแฟ้มเก่าจะถูกเก็บในแฟ้มสำรองชื่อ file.BAK ในที่นี้เก็บใน TEST.BAK (ถ้าสิ่งที่เก็บในแฟ้ม TEST.FOR หายไป เราสามารถเรียกแฟ้ม TEST.BAK มาแก้ไขใหม่ได้)

การปฏิบัติเมื่อแฟ้ม TEST.FOR หายไป

1. ตรวจสอบข้อความในแฟ้ม TEST BAK ก่อน ว่ามีสิ่งที่ต้องการหรือไม่ โดยใช้

คำสั่ง A>TYPE TEST.BAK (E) (ถ้าใช้ช่องรับ A อยู่)

ถ้ามีสิ่งที่ต้องการให้ทำข้อ 2 ต่อไป ถ้าใน TEST.BAK ไม่มีข้อความที่ต้องการ นักศึกษาต้องพิมพ์ข้อความหรือโปรแกรมใหม่ทั้งหมด

2. ตรวจสอบดูว่าในแผ่นดิสเกตต์ยังมีแฟ้มชื่อ TEST.FOR อยู่หรือไม่โดยใช้คำสั่ง

คำสั่ง A>DIR (E)

3. ทำการลบแฟ้ม ออกจากแผ่นดิสเกตต์เสียก่อน โดยใช้คำสั่ง

คำสั่ง A>DEL TEST.FOR (E)

4. ทำการทำสำเนาแฟ้มจากแฟ้ม TEST.BAK ไปเก็บในแฟ้ม TEST.FOR โดยใช้คำสั่ง

คำสั่ง A>COPY TEST.BAK TEST.FOR (E)

5. ลองพิมพ์ข้อความในแฟ้ม TEST.FOR ออกมาดูโดยใช้คำสั่ง

คำสั่ง A>TYPE TEST.FOR (E)

ในแฟ้ม TEST-FOR จะมีข้อความเหมือน TEST.BAK

ขั้นที่ 2

คำสั่ง B>A: PAS2 (E)

ถ้าไม่มีข้อผิดพลาดในขั้นนี้ จะได้ข้อความบนจอในลักษณะดังนี้ (เนื้อที่ที่ใช้จะแตกต่างกันในแต่ละโปรแกรม)

Code Area Size = #05EC (1516) เลขฐาน 16 เลขฐาน 10
 Cons Area Size = #00E6 (230)
 Data Area Size = #0264 (612)

Pass Two No Errors Detected.

ข้อความนี้บอกให้ทราบขนาด (จำนวนไบต์) ของที่ที่ใช้เก็บคำสั่งที่คอมพิวเตอร์จะปฏิบัติตามได้ (Code) ค่าคงที่ (Constant) ข้อมูลหรือตัวแปร (Data หรือ Variables) ตามลำดับ

ภายหลังขั้นที่ 2 นี้ ถ้าคุณเพิ่มใน directory ในช่องขับ B จะเห็นว่า มีแฟ้ม TEST.OBJ เพิ่มขึ้น แฟ้ม TEST.OBJ เป็นแฟ้มที่เก็บโปรแกรมในภาษาเครื่อง (object program) ทั้งนี้คอมไพเลอร์จะสร้างแฟ้ม TEST.OBJ ขึ้นก็ต่อเมื่อไม่มี compile-time error ในแฟ้ม TEST.FOR (ในกรณีที่มี error เป็นเพียง warning (เตือน) คอมไพเลอร์จะสร้างแฟ้ม TEST.OBJ ให้ แต่เมื่อวิ่งโปรแกรมอาจเกิดข้อผิดพลาดได้ ดังนั้นควรแก้ไขข้อผิดพลาดให้หมด)

ขั้นที่ 3 ขั้นทำการ Link โปรแกรม

เอาคอมไพเลอร์แผ่นที่ 2 ใส่ในช่องขับ A แทนแผ่นที่ 1

คำสั่ง B>A:LINK (E)

โปรแกรม linker จะถามคำถาม 4 คำถาม โดยถามทีละคำถามให้ตอบดังนี้

Object modules[.OBJ]:TEST (E) ---> ให้เอาแฟ้ม TEST.OBJ จาก
 แผ่นดิสเกตต์ในช่องขับ B
 Ruh file [TEST.EXE]: (E) ----> สวมจับชื่อ TEST.EXE
 List file[NUL.MAP]: (E)
 LibrariesC.LIB:A: (E) ---> ใช้ libraries จากแผ่นดิสเกตต์ใน
 ช่องขับ A

รอนจนได้ prompt sign ถ้ามีข้อผิดพลาดในการ link เกิดขึ้น จะมีข้อความบอก
บนจอ ต้องแก้ไขก่อน แล้ว link ใหม่ จนไม่มีข้อผิดพลาด จึงทำขั้นที่ 4 ต่อไป

ลองตรวจดูแฟ้มในแผ่นดิสเกตต์ในช่องขับ B จะเห็นแฟ้ม TEST.ext ที่มี .ext
เป็น .FOR, .DAT, .LST, .OBJ, .EXE

ขั้นที่ 4 วิ่งโปรแกรม (run program)

คำสั่ง B>TEST **(E)**

สั่งให้ execute คำสั่งในแฟ้ม TEST.EXE

ถ้าคอมพิวเตอร์วิ่งโปรแกรมจนจบ (ไม่มีปัญหาใด ๆ เกิดขึ้น) จะได้ข้อความบนจอภาพ
ดังนี้

Stop - Program Terminated.

ถ้าในโปรแกรมมีคำสั่ง OPEN เพื่อเปิดแฟ้ม TEST.OUT ไว้สำหรับเก็บข้อมูลออก
ในแผ่นดิสเกตต์ในช่องขับ B จะมีแฟ้ม TEST.OUT เกิดขึ้นด้วย ดูได้จากใน directory

ขั้นที่ 5 ตรวจสอบ output data ใน output file โดยใช้คำสั่ง

คำสั่ง B>TYPE TEST-OUT **(E)**

ถ้าคำตอบผิดซึ่งอาจเนื่องมาจากตรรกะ (logic) ของการสั่งงานผิด
(โปรแกรมผิด) หรืออาจเนื่องมาจาก input data ผิด หรือผิดทั้งคู่ ให้ปฏิบัติดังนี้

- 1) ถ้าโปรแกรมผิด แก้ในแฟ้ม TEST.FOR แล้วทำขั้นที่ 1 ใหม่
- 2) ถ้า data ผิด (โปรแกรมไม่ผิด) แก้ในแฟ้ม TEST.DAT แล้วทำขั้นที่ 4
ได้ทันที
- 3) ถ้าทั้งโปรแกรมและ data ผิด แก้ในแฟ้ม TEST.FOR และ TEST.DAT
แล้วไปทำขั้นที่ 1 ใหม่

4.2 เมื่อ Microsoft FORTRAN Compiler V3.31 อยู่ใน Hard disk

ขั้นตอนการจัดเก็บ (install) compiler (ซึ่งเดิมเก็บอยู่ในแผ่นดิสเกตต์ 2 แผ่น) ไว้ใน Hard disk

ขั้นที่ 1 ขณะที่อยู่ใน root directory ให้ DOS command MD (Make Directory) เพื่อสร้าง subdirectory ชื่อ FORTRAN ไว้ก่อนดังนี้

คำสั่ง C:\>MD FORTRAN (E)

จะมี subdirectory FORTRAN เกิดขึ้น

ต่อไปใช้คำสั่ง CD (Change Directory) เพื่อเข้าไปใน subdirectory FORTRAN ดังนี้

คำสั่ง C:\>CD FORTRAN (E)

จะได้ prompt sign คือ C:\FORTRAN>

ต่อไปทำการ copy compiler จากแผ่นดิสเกตต์เข้าไปเก็บใน subdirectory FORTRAN

ใส่แผ่นที่ 1 ลงในช่องขับ A

คำสั่ง C:\FORTRAN>COPY A:*. * (E)

รจนจะได้ C:\FORTRAN>

ต่อไปใส่แผ่นที่ 2 ลงในช่องขับ A กด F3 เรียกคำสั่ง COPY เดิมออกมา กด (E)
เมื่อ copy เสร็จแล้ว ลองตรวจสอบใน subdirectory FORTRAN ดู

คำสั่ง C:\FORTRAN>DIR (E)

จะเห็นแฟ้ม FOR1, PAS2, LINK, etc. อยู่

ขั้นตอนการคอมไพล์

ขั้นเตรียม ใส่แผ่นที่มี file TEST.FOR ซึ่งเก็บ source program และมี file TEST.DAT ซึ่งเก็บข้อมูลเข้า ลงในช่องขับ A

ขั้นที่ 1 เริ่มทำขั้นที่ 1 ของการคอมไพล์ โดยใช้คำสั่ง

คำสั่ง C:\FORTRAN>FOR1(E)

คอมไพเลอร์จะถามคำถาม 4 คำถาม (ตัวอย่างคำตอบคือที่ขีดเส้นใต้)

Source filename [.FOR]: A:TEST(E) ---> คอมไพเลอร์จะหาโปรแกรมใน
แฟ้ม TEST.FOR ในช่องขับ A
Object filename [TEST.OBJ]: A:(E) ---> ขอรับชื่อ TEST.OBJ แต่เก็บ
ในช่องขับ A (ถ้าไม่ใช่ A: TEST.OBJ จะถูกเก็บใน harddisk)
Source listing [NUL.LST]: A:TEST(E) ---> เพื่อให้คอมไพเลอร์สร้าง
แฟ้ม TEST.LST เก็บในแผ่นดิสเกตต์ในช่องขับ A
Object listing [NUL.COD]: (E) ---> ไม่เก็บ Object listing

คอมไพเลอร์จะทำการคอมไพล์โปรแกรมในแฟ้ม TEST.FOR

ถ้าโปรแกรมไม่มี Compile-time error บนจอภาพจะปรากฏข้อความว่า

Pass one No Errors Detected.

ถ้าจอภาพปรากฏข้อความ เช่น

Pass one 3 Errors Detected.



จะบอกสิ่งที่มี 3 แห่งที่คอมไพเลอร์ตรวจพบ นักศึกษาจะดูสิ่งที่มีในคำสั่งที่พิมพ์ได้โดยดูในแฟ้ม TEST.LST ซึ่งถูกบันทึกในแผ่นดิสเกตต์ในช่องขับ A โดยใช้คำสั่งต่อไปนี้

คำสั่ง C:\FORTRAN>TYPE A:TEST.LST (E)

(ใช้ Ctrl-S หยุดจอภาพ และกดปุ่มใด ๆ เพื่อพิมพ์ต่อ)

ให้นักศึกษานำโปรแกรมในแฟ้ม TEST.FOR โดยใช้ WordStar เมื่อกดแป้นเสร็จแล้ว
กลับไปทำขั้นที่ 1 ซ้ำ จนได้ข้อความบนจอว่า Pass one No Errors Detected.
แล้วทำขั้นที่ 2 ต่อ

ขั้นที่ 2

คำสั่ง C:\FORTRAN>PAS2 (E)

ถ้าไม่มีข้อผิดพลาดในขั้นนี้ จะได้ข้อความบนจอในลักษณะดังนี้ (เนื้อที่ที่ใช้จะแตกต่างกันใน
แต่ละโปรแกรม)

Code Area Size = #05EC (1516) เลขฐาน 16 เลขฐาน 10
Cons Area Size = #00E6 (230)
Data Area Size = #0264 (612)

Pass Two No Errors Detected.

ข้อความนี้บอกให้ทราบขนาด (จำนวนไบต์) ของที่ที่ใช้เก็บคำสั่งที่คอมพิวเตอร์จะปฏิบัติตาม
ได้ (Code) ค่าคงที่ (Constant) ข้อมูลหรือตัวแปร (Data หรือ Variables) ตามลำดับ
ภายหลังขั้นที่ 2 นี้ ถ้าคุณเพิ่มใน directory ในช่องขับ A จะเห็นว่าแฟ้ม TEST.OBJ
เพิ่มขึ้น

ขั้นที่ 3 ทำการ Link โปรแกรม

คำสั่ง C:\FORTRAN>LINK (E)

โปรแกรม linker จะถามคำถาม 4 คำถาม โดยถามทีละคำถามให้ตอบดังนี้

Object modules[.OBJ]:A:TEST (E) ---> ให้เอาแฟ้ม TEST.OBJ จาก
แผ่นดิสเกตต์ในช่องขับ A

Run file [C:TEST.EXE]:A:(E) ---> สอนรับชื่อ TEST.EXE แต่ให้
เก็บในแผ่นดิสเกตต์ในช่องขับ A

List file[NUL.MAP]:(E)

Libraries[.LIB]:(E) ---> ให้ libraries จาก hard disk

รจนได้ prompt sign ถ้ามีข้อผิดพลาดในการ link เกิดขึ้น จะมีข้อความ
บอกบนจอ ต้องแก้ไขก่อน แล้ว link ใหม่จนไม่มีข้อผิดพลาด จึงทำขั้นที่ 4 ต่อไป

ขั้นที่ 4 วิ่งโปรแกรม (run program)

คำสั่ง C:\FORTRAN>A:TEST (E)

สั่งให้ execute คำสั่งในแฟ้ม TEST.EXE

ถ้าคอมพิวเตอร์วิ่งโปรแกรมจนจบ (ไม่มีปัญหาใด ๆ เกิดขึ้น) จะได้ข้อความบนจอภาพ

ดังนี้

stop - Program Terminated.

ถ้าในโปรแกรมมีคำสั่ง OPEN เพื่อเปิดแฟ้ม TEST.OUT ไว้สำหรับเก็บข้อมูลออก
ในแผ่นดิสเกตต์ในช่องขับ A จะมีแฟ้ม TEST.OUT เกิดขึ้นด้วย คุ้ได้จากใน directory

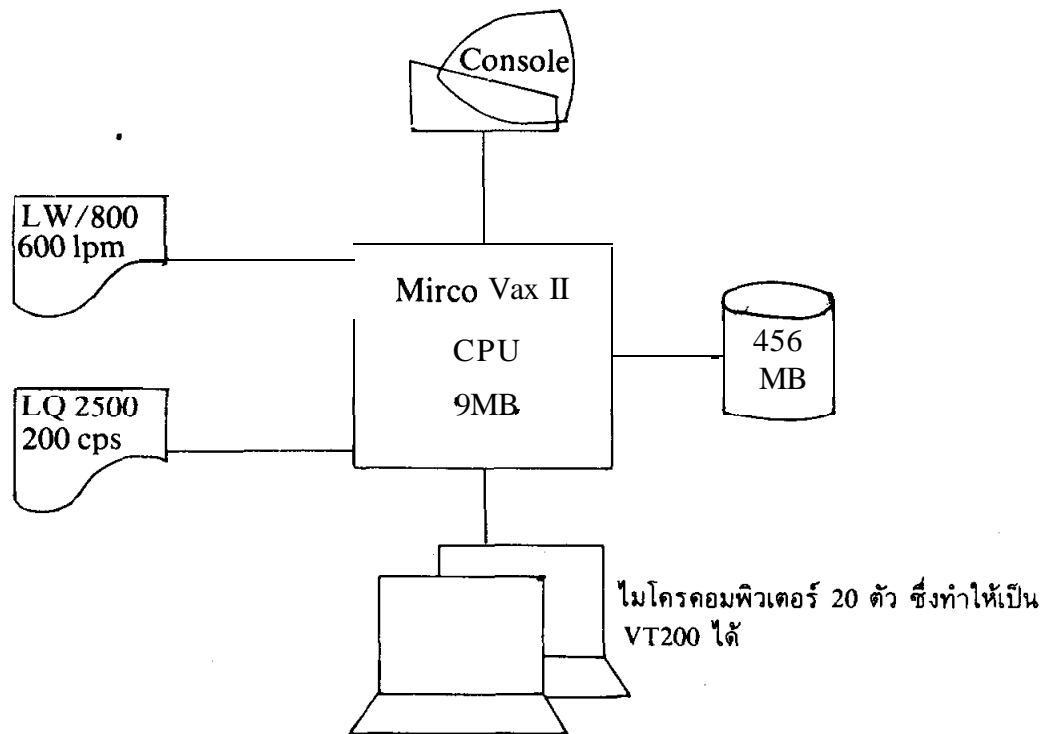
ขั้นที่ 5 ตรวจสอบ output data ใน output file โดยใช้คำสั่ง

คำสั่ง C:\FORTRAN>TYPE A:TEST.OUT (E)

ถ้าคำตอบผิดซึ่งอาจเนื่องมาจากตรรกะของการสั่งงานผิด (โปรแกรมผิด) หรืออาจเนื่องมาจาก input data ผิด หรือผิดทั้งคู่ ให้ปฏิบัติดังนี้

- 1) ถ้าโปรแกรมผิด แก้ในแฟ้ม TEST.FOR แล้วทำขั้นที่ 1 ใหม่
- 2) ถ้า data ผิด (โปรแกรมไม่ผิด) แก้ในแฟ้ม TEST.DAT แล้วทำขั้นที่ 4 ได้ทันที
- 3) ถ้าทั้งโปรแกรมและ data ผิด แก้ในแฟ้ม TEST.FOR และ TEST.DAT แล้วไปทำขั้นที่ 1 ใหม่

5. ระบบคอมพิวเตอร์ Hicro VAX II



รูปแสดงส่วนประกอบของระบบ Micro VAX II

5.1 ความรู้ทั่วไปเกี่ยวกับระบบ Hicro VAX II

Hicro VAX II เป็นเครื่องของ DEC (Digital Equipment Corporation) จัดเป็นซูเปอร์ไมโครคอมพิวเตอร์ (Super microcomputer)

หน่วยความจำ ประกอบด้วย

หน่วยความจำหลักขนาดความจุ 9 MB

หน่วยความจำสำรองเป็น Fixed disk ความจุ 456 MB

หน่วยความจำ Back up เป็น Cartridge tape

ความจุ 95 MB ต่อม้วน

Operating System และ Version

Micro VMS V4.5 (VMS = Vertual Memory System)

หน้าจอนำข้อมูลเข้า

ไมโครคอมพิวเตอร์ (เป็น IBM Compatible) 20 เครื่อง
ซึ่งทำหน้าที่เลียนแบบ (emulate) เทอร์มินัล series 200 (VT 200: Video terminal series 200) ได้

หน้าจอนำข้อมูลออก

เครื่องพิมพ์ที่ละบรรทัด (Line printer) 1 เครื่อง คือเครื่อง LW/800
ความเร็วในการพิมพ์ 600 บรรทัด/นาที (lpm = line per minute)

เครื่องพิมพ์ดอตเมตริกซ์ 1 เครื่อง คือ LQ2500 ความเร็วในการพิมพ์
200 ตัวอักษร/วินาที (cps = character per second)

ภาษาฟอร์แทรน

ภาษาฟอร์แทรนที่ใช้คือ VAX-FORTRAN V4.4-177

5.2 เมื่อเลียนแบบ (Emulate) ไมโครคอมพิวเตอร์เป็น VT Series 200

เมื่อเลียนแบบไมโครคอมพิวเตอร์เป็น VT series 200 ปุ่ม Enter หรือ **(E)**
แทน Return

Function keys F1-F10 จะทำหน้าที่ดังต่อไปนี้ คือ

F1

ทำหน้าที่แทน PF1 ใช้ปุ่มนี้ร่วมกับปุ่มอื่น เช่นกด PF1 (แล้วปล่อย) ตามด้วย PF4 (กด F1 ตามด้วย F4) จะได้ UND LINE (นำเอาบรรทัดที่เก็บอยู่ใน line buffer ออกมาบนหน้าจอ ณ ตำแหน่งที่เคอร์เซอร์อยู่)

F2

ทำหน้าที่แทน PF2 กดปุ่มนี้เพื่อดูคำอธิบายรายละเอียดในการใช้ปุ่มต่าง ๆ คือ HELP

F3

ทำหน้าที่แทน PF3

F4

ทำหน้าที่แทน PF4 ใช้ปุ่มนี้เพื่อลบบรรทัด (DEL LINE) ในการลบทั้งบรรทัด ให้ตั้งเคอร์เซอร์ที่ต้นบรรทัด แล้วกด F4
ถ้าต้องการให้บรรทัดที่เพิ่งถูกลบไปกลับมาบนจอภาพ ณ ตำแหน่งที่เคอร์เซอร์อยู่ ให้กด F1 ตามด้วย F4

'F5 - F8 ใช้ในการเลื่อนเคอร์เซอร์เมื่ออยู่ใน editor ใน keypad mode

F5	ทำหน้าทีนทน	↑
F6	ทำหน้าทีนทน	↓
F7	ทำหน้าทีนทน	←
F8	ทำหน้าทีนทน	→
F9	ทำหน้าทีนทน Hold Screen	เมื่อใช้ DCL Command TYPE ใช้เพื่อให้หยุด/ให้พิมพ์ต่อไปบนจอภาพ ในการหยุดจอภาพอาจใช้ Ctrl-S ถ้าหยุดด้วย Ctrl-S ต้องใช้ <u>Ctrl-Q</u> เพื่อพิมพ์ต่อ
F10	ทำหน้าทีนทนปุ่ม Line feed	

คำสั่ง **DCL (Digital Control Language)** ที่เราใช้ทุกคำสั่งในขณะที่ log on อยู่ จะถูกเก็บไว้ใน buffer ถ้าเราต้องการใช้คำสั่งบางคำสั่งที่เคยใช้แล้ว นักศึกษาอาจเรียกมาใช้ได้โดยไม่ต้องพิมพ์คำสั่งนั้นใหม่ การเรียกจาก buffer ทำได้โดยกดปุ่ม F5 และ F6 เพื่อค้นหาสิ่งที่ต้องการ เมื่อ log off แล้วคำสั่ง DCL ใน buffer จะหายไปหมด (buffer คือที่เก็บชั่วคราวของเทอร์มินัล)

หมายเหตุ ให้นักศึกษาอ่านตอน 2 ข้อ 3 เป็นต้นไปประกอบด้วย

5.3 ขั้นตอนการ Log On เครื่อง Micro VAX II

ขั้นเตรียม ใส่แผ่นดิสก์เกตต์ที่มีโปรแกรม Poly-COM ในช่องขับ A

ถ้ายังไม่ได้เปิดเครื่อง ให้เปิดจอภาพ เปิดที่พืษุ เครื่องจะบูท DOS ในแผ่น Poly-COM เข้าไปในหน่วยความจำหลัก ให้ใส่วัน เวลา แล้วรอนได้ A>

ถ้าเปิดเครื่องอยู่แล้วและอยู่ที่ A> ให้ทำขั้นที่ 1

ขั้นที่ 1 เร็กใช้ Poly-COM โดยใช้คำสั่ง

คำสั่ง A>TRM(E)

รอนได้ block cursor (█) บนจอภาพ แล้วทำขั้นที่ 2 ต่อไป

ถ้าเปลี่ยนใจจะไม่ log on เครื่อง Micro VAX II ให้กด [Alt] แะไว้ แล้วกด [F10] ปล่อขทังคู้ แล้วกด [X] ตาม จะได้ข้อความบนจอภาพดังนี้

poly-COM/240 exit

A>

ขั้นที่ 2 กด (E) บนจอจะปรากฏข้อความ 3 บรรทัด และถาม Username ของผู้ใช้ (Username คือชื่อ account ของนักศึกษา ซึ่งจะให้แก่ักศึกษาแต่ละคนที่เข้าเรียน)

Micro VAX II
Micro VAX II
Micro VMS v4.5

Username: พิมพ์ username **E** เช่น A1 **E**

Password: พิมพ์ password **E** สำหรับการ log on ครั้งแรกใช้ password
ตัวเดียวกับ username เมื่อพิมพ์ password
สิ่งที่พิมพ์จะไม่ปรากฏบนจอภาพ ทั้งนี้เพื่อเป็นการ
ป้องกันไม่ให้ผู้อื่นเข้าถึงข้อมูลใน account

ถ้าทำไม่ถูกต้อง คือไม่ได้เครื่องหมาย \$ (DCL prompt) บนจอ ให้กด **E** อีก
เครื่องจะถาม username อีก แล้วถาม password

ถ้าทำถูกต้องบนจอจะปรากฏข้อความว่า

Welcome to Micro VMS v4.5

Your terminal is:TXA1

\$

↑
ชื่อจะต่างกันไปตามเครื่องที่ใช้คือ TXA0, TXA1, . . .
TXA7, TXB4, . . ., TXB7, TXCO, . . ., TXC7

ให้นักศึกษาจดชื่อเทอร์มินัลที่ตนใช้ไว้
เทอร์มินัลที่ใช้ชื่อ _____

เครื่องหมาย \$ แสดงว่าระบบพร้อมที่จะรับคำสั่ง DCL แล้ว (DCL command =
Digital control language command) ทั้งนี้จะรับคำสั่งได้ทีละคำสั่ง
คำสั่ง DCL เช่น EDIT, DIR, COPY, TYPE, DEL, PURGE เป็นต้น

ขั้นที่ 3 ให้ทำการเปลี่ยน password ใหม่ โดยใช้คำสั่ง

คำสั่ง \$SET PASS (E)

จะได้ข้อความบนจอให้ตอบ

Old password:: พิมพ์ password เก่า (E)

New password: พิมพ์ password ใหม่ (E) } ต้องเหมือนกัน

ตรวจทาน --> Verification: พิมพ์ password ใหม่ (E)

\$ (ถ้าการตั้ง password ใหม่เรียบร้อยจะได้ prospt sign แต่

ถ้ายังไม่ถูกต้อง ให้กลับไปใช้คำสั่ง SET PASS ใหม่)

หมายเหตุ password ต้องเป็นคำที่ยาวอย่างน้อย 6 ตัวอักษร ใช้ตัวเลขปนตัวอักษรได้
ในขณะที่พิมพ์ password ทั้ง 3 ครั้ง สิ่งพิมพ์จะไม่ปรากฏบนจอ ทั้งนี้เพื่อ
ความปลอดภัยของ account และเพื่อเป็นการตรวจสอบ password ที่ตั้ง
ใหม่ ให้ทำการ log off โดยใช้คำสั่ง

คำสั่ง \$LOG (E)

บนจอภาพจะปรากฏข้อความบอกว่า log off เมื่อวันเวลาใด จากนั้นให้ทำการ
log on ใหม่ โดยกลับไปทำขั้นที่ 2 การ log on ครั้งนี้และต่อ ๆ ไปต้องใช้
password ใหม่

5.4 การส่งสิ่งที่เก็บในแฟ้มในแผ่นดิสเกตต์ไปเก็บใน Account

ใน Micro VAX II

ขั้นเตรียมการ

ให้ log on เครื่อง Micro VAX II จนได้ \$ บนจอ เอานแผ่นดิสเกตต์
ซึ่งมีแฟ้มที่จะส่งข้อความไปเก็บใน Micro VAX II (คือเก็บใน fixed disk) ใส่ใน
ช่องขับ A

ขั้นที่ 1 เริ่มทำการ copy เพิ่ม โดยใช้คำสั่ง

คำสั่ง `SCOPY` (E)

ระบบจะถามว่าจะ copy จากแผ่นใด ไปเก็บในแผ่นใด

-From:TXA1:file.ext (E)

TXA1 คือชื่อเทอร์มินัลที่กำลังใช้งานอยู่
file.ext คือชื่อแผ่นในแผ่นดิสเกตต์ที่จะ copy หรือความไปเก็บใน
Micro VAX II

-To:file.ext (E)

file.ext คือชื่อแผ่นที่จะเรียกใน Micro VAX II อาจใช้ชื่อเก่าหรือชื่อใหม่
ก็ได้ (ในที่นี้ให้ใช้ชื่อเก่าเหมือนกับที่ใช้เก็บในแผ่นดิสเกตต์)

ทั้งนี้อาจใช้คำสั่งต่อไปนี้แทนได้

คำสั่ง `SCOPY TXA1:file.ext file.ext` (E)

ขั้นที่ 2 ทำการส่งเพิ่ม (Send file = S)

กดปุ่ม Alt แล้วย้ำก่อน กด F10 แล้วปล่อยทั้งคู่ กด S ตาม
เครื่องจะถามชื่อแผ่นที่ต้องการส่งไปยัง Micro VAX II

Send file:file.ext (E) (ถ้าแผ่นดิสเกตต์ที่มี file.ext อยู่ในช่องขับ A)

หรือ Send file:B:file.ext (E) (ถ้าแผ่นดิสเกตต์ที่มี file.ext อยู่ในช่องขับ B)

ถ้าการส่งข้อมูลไปยัง Micro VAX II ถูกต้อง จะปรากฏสิ่งที่อยู่ในแฟ้มที่ถูกส่ง
บนจอ เครื่องจะแสดงไปจนจบแฟ้มแล้วหยุด (ถ้าสิ่งที่ปรากฏบนจอไม่เหมือนกับในแฟ้ม
คือมีข้อความอื่นปะปนอยู่ด้วย แสดงว่าการส่งไม่สำเร็จ จะต้องกลับไปเริ่มต้นที่ขั้นที่ 1
ใหม่)

ขั้นที่ 3 กด Ctrl-Z (^Z) จะได้ \$ บนจอ แสดงว่าแฟ้มได้ถูกจัดเก็บแล้ว

ถ้าต้องการส่งแฟ้มอื่นไปเก็บใน Micro VAX II อีก ให้กลับไปทำขั้นที่ 1 อีก
การส่งส่งได้ที่ละแฟ้มเท่านั้น ถ้าจำชื่อเทอร์มินัลไม่ได้ให้ใช้คำสั่ง

คำสั่ง \$SHOW TERM **(E)**
(TERM = terminal)

ขั้นที่ 4 ลองตรวจดูรายชื่อแฟ้มใน account

คำสั่ง \$DIR **(E)**

เครื่องจะแสดง directory ใน account ของนักศึกษาใน Micro VAX II

ก่อนจะทำคำสั่งอื่นต่อไป เราอาจตรวจสอบสิ่งที่เก็บในแฟ้มที่รับข้อความมาจาก
ไมโครคอมพิวเตอร์ โดยใช้คำสั่ง TYPE คำสั่งนี้จะนำเอาสิ่งที่เก็บในแฟ้มที่ระบุออกมา
แสดงที่หน้าจอ โดยจะพิมพ์ไปเรื่อย ๆ จนจบแฟ้ม เรากด **F9** เพื่อให้หยุดจอภาพหรือ
เพื่อพิมพ์ต่อ

คำสั่ง \$TYPE file.ext(;version ที่) **(E)**

ถ้าไม่ระบุ version ที่เครื่องจะเอาสิ่งที่เก็บใน file.ext version ล่าสุดออก
มาแสดงบนจอ

5.5 การรับแฟ้มใน Micro VAX II มาเก็บในแผ่นดิสเกตต์

ขั้นเตรียมการ

log on กับ Micro VAX II แล้วใส่แผ่นดิสเกตต์ที่จะใช้บันทึกแฟ้มในช่องขับ A [สมมุติว่าแฟ้มใน Micro VAX II ชื่อ NEW.FOR ให้แฟ้มในแผ่นดิสเกตต์ (ที่จะเก็บข้อความจาก NEW.FOR) ชื่อ NNEW.FOR]

ขั้นที่ 1 ใช้คำสั่ง COPY แฟ้ม เช่นเดียวกับในข้อ 5.4

```

คำสั่ง      $COPY (E)
            -From:NEW.FOR (E)
            -To: กด [Alt] แล้วย้ำก่อน แล้วกด [F10] ตาม แล้วปล่อยทั้งคู่
                ต่อไปกด [R] (R = receive)
บนจอภาพจะปรากฏข้อความถามชื่อแฟ้ม
Receive file:NNEW.FOR (E) บนจอจะปรากฏข้อความเดิมตั้งแต่
                                COPY ถึง -To: ให้พิมพ์ชื่อเทปมินิลที่ใช้อยู่
            -To:TXA1: (E)

```

เครื่องจะพิมพ์ข้อความในแฟ้มที่รับจาก Micro VAX II บนจอภาพจนจบแฟ้ม และจะ
ได้ \$ บนจอ

ขั้นที่ 2 กด [Alt] [F10] [X] กดวิธีเดียวกับในขั้นที่ 1

```

บนจอจะปรากฏข้อความ
poly-COM/240 exit
A>

```

ตรวจสอบสารบาณแฟ้มในแผ่นดิสเกตต์ในช่องขับ A โดยใช้คำสั่ง

```

คำสั่ง      A>DIR (E)

```


จะเห็นว่ามีแฟ้ม NNEW.FOR ในแผ่นดิสเกตต์ ต่อไปลองพิมพ์ข้อความในแฟ้ม NNEW.FOR ออกมาดูที่จอภาพ

คำสั่ง A>TYPE NNEW.FOR (E)

เครื่องจะพิมพ์ข้อความออกมาที่จอภาพ และได้เครื่องหมาย \$ เกินมาด้วยที่ท้ายแฟ้ม ให้ใช้ WordStar ลบเครื่องหมาย \$ ออกก่อนนำไปใช้ต่อไป

5.6 การ Log Off จาก Micro VAX II

ทุกครั้งที่เลิกใช้ Micro VAX II แล้วจะใช้ไมโครคอมพิวเตอร์ต่อ ให้ log off จาก Micro VAX II ทุกครั้ง ทั้งนี้เพื่อกันลืมการ log off จาก account ของตน

คำสั่ง \$LO (E)

จะได้ข้อความทำนองต่อไปนี้

```
Al logged out at 6-JULY-1967 10:59:30.35
  ^
  |
  | username
```

ต่อไปให้ออกจาก poly-cow โดยการกด Alt, F10 และ X จะได้ข้อความว่า

```
poly-COM/240 exit
A>
```

หมายเหตุ ถ้านักศึกษาไม่ได้ log off จาก Micro VAX II แล้วออกจาก poly-COM มาใช้ไมโครคอมพิวเตอร์ ต่อไปต้องการกลับไปใช้ Micro VAX II อีก ทำได้โดยเอาแผ่น poly-COM ใสในช่องขับ A แล้วใช้คำสั่ง

คำสั่ง A>TRM (E)

เมื่อได้ block cursor บนจอภาพ ให้กด (E) ถ้าได้ \$ ให้ทำงานต่อไปได้
ถ้าไม่ได้ \$ บนจอให้กด Ctrl-Q หรือ Ctrl-Y แล้วรอจนได้ \$ บนจอ

5.7 คำสั่งพิมพ์แฟ้มด้วยเครื่องพิมพ์ที่ละบรรทัด (Line Printer)

เครื่องพิมพ์ LW/800 เป็นเครื่องพิมพ์แคร์สาว กระดาษต่อเนื่องใช้พิมพ์ได้บรรทัด
ละ 132 ตัวอักษร

คำสั่ง \$PRINT file1.ext (, file2.ext, file3.ext,...)

ถ้าต้องการพิมพ์มากกว่า 1 แฟ้ม ให้ใช้เครื่องหมาย , คั่นระหว่างชื่อแฟ้ม อาจ
ระบุ version ของแฟ้มที่ต้องการได้ด้วย ถ้าไม่ระบุ version เครื่องจะเอา
version สุดท้ายออกมาพิมพ์ให้

6. ตัวอย่างการปฏิบัติงาน (Sample Session)

ให้นักศึกษาใช้ WordStar ในการสร้างแฟ้ม TEST.FOR เพื่อเก็บ FORTRAN
source program และสร้างแฟ้ม TEST.DAT เพื่อเก็บข้อมูลเข้า 2 เรคคอร์ด
ต่อจากนั้นให้ใช้ Micro VAX II ในการคอมไพล์และวิ่งโปรแกรม

กำหนด Input record 2 records เพื่อเก็บในแฟ้ม TEST.DAT

column ที่	1234567890...
เรคคอร์ดที่ 1	4.0 9.7
เรคคอร์ดที่ 2	8.0 4.5

กำหนด FORTRAN source program

```
c2345 | 78901234567890...
C | THE FIRST PROGRAM IN OR 223
C | OPEN INPUT FILE (OLD FILE) AND OUTPUT FILE (NEW FILE)
C | OPEN(11,FILE='TEST.DAT',STATUS='OLD')
C | OPEN(12,FILE='TEST.OUT',STATUS='NEW')
C |
C | READ THE FIRST RECORD FROM INPUT FILE TEST.DAT
C | READ(11,3)A,B
3 | FORMAT(F3.1,F4.1)
C | AVER = ( A + B ) / 2.0
C |
C | WRITE AVERAGE OF A AND B TO OUTPUT FILE TEST.OUT
C | WRITE(12,4)AVER
4 | FORMAT(1X,'AVERAGE OF A AND B IS',F5.2)
C |
C | READ THE NEXT RECORD FROM INPUT FILE TEST.DAT
C | READ(11,3)C,D
C | DIFF = D - C
C | WRITE THE DIFFERENCE BETWEEN C AND D
C | TO OUTPUT FILE TEST.OUT
C | WRITE(12,5)DIFF
5 | FORMAT(1X,'DIFFERENCE BETWEEN C AND D IS',F5.2)
C | CLOSE FILES
C | CLOSE(11)
C | CLOSE(12)
C | STOP
C | END
```

ให้นักศึกษาปฏิบัติตามขั้นตอนต่อไปนี้

6.1 ใส่แผ่น WordStar ในช่องขับ A ปิดลิ้นเปิดจอภาพ เปิดรีพีต

6.2 ใส่วันเวลาให้ถูกต้อง จะได้ A> บนจอ (ดูหัวข้อ 2.3)

6.3 A>WS(E) รจนได้ Opening menu

6.4 กด N เพื่อเปิด non-document file

เมื่อถูกถามชื่อแฟ้ม ให้พิมพ์ TEST.FOR(E) รจนได้ Main menu เคอร์เซอร์จะอยู่ที่คอลัมน์ที่ 1 บรรทัดที่ 1 (FC = 1, FL = 1, COL = 1) ซึ่งเป็นตำแหน่งแรกของแฟ้ม ให้นักศึกษาพิมพ์โปรแกรมเหมือนกับการเขียนโปรแกรมลง coding sheet ดู indicator (FC, FL, COL) ที่บรรทัดแรกของเมนูประกอบได้

ก่อนพิมพ์ให้กดปุ่ม Caps Lock ไว้ก่อน (จะใช้อักษรตัวใหญ่ในการพิมพ์)

กด Ctrl-V (ค่า INSERT ON จะหายไปจากบรรทัดแรก เพื่อให้แทรกข้อความขณะพิมพ์)

เมื่อพิมพ์จบแต่ละบรรทัดกด(E)เคอร์เซอร์จะลงมาอยู่ที่คอลัมน์ที่ 1 ของบรรทัดถัดไป ให้พิมพ์โปรแกรมจนถึงคำสั่ง END แล้วต้องกด(E)

6.5 กด Ctrl-K จะได้ Block menu ให้กด D (^KD = ให้ save ข้อความที่พิมพ์ทั้งหมดในแฟ้ม TEST.FOR แฟ้มนี้จะถูกเก็บในแผ่นดิสเกตต์ในช่องขับ A เมื่อ save แฟ้มนี้แล้ว เรายังคงอยู่ใน WordStar) จะปรากฏข้อความว่ากำลัง save แฟ้มอยู่ แล้วจะปรากฏ Opening menu ให้ดูใน directory ได้เมนู จะเห็นแฟ้ม TEST.FOR ด้วย

6.6 ต่อไปสร้างแฟ้ม TEST.DAT สำหรับเก็บข้อมูลเข้า 2 เรคคอร์ด

กด N

ใส่ชื่อแฟ้ม TEST.DAT(E) (เมื่อถูกถาม) เมื่อได้ Main menu ให้เริ่มพิมพ์ตั้งแต่คอลัมน์ที่ 1 บรรทัดที่ 1

4.0 9.7 0 E

6.0 4.6 0 E

6.7 กด Ctrl-K แล้วกด X เมื่อไปอยู่ที่ Block menu (^KX คือให้เก็บ 2 เรคคอร์ด ในแฟ้ม TEST.DAT แล้วออกจาก WordStar ไปยังระบบ) จะได้ A> บนจอ

6.8 A>DIR (E) เพื่อดูรายชื่อแฟ้ม เนื้อหาในแต่ละแฟ้ม วันเวลาที่สร้างแฟ้ม หรือ A>DIR/W เพื่อดูรายชื่อแฟ้มในแนวขวางซึ่งจะแสดงแต่รายชื่อแฟ้มเท่านั้น

6.9 ทำการ log on Micro VAX II แล้วส่งแฟ้ม TEST.FOR และ แฟ้ม TEST.DAT เข้าไปเก็บใน account ของนักศึกษา ให้ทำตามลำดับดังนี้

6.9.1 ใส่แผ่น Poly-COM ในช่องขับ A

6.9.2 A>TRM (E) รอจนได้ block cursor บนจอ แล้วกด (E)

6.9.3 ใส่ Username และ Password ที่กำหนดให้นักศึกษาแต่ละคน จดชื่อ e-mail ของตนไว้ ในที่นี้สมมติว่าเป็น TXA1

ต่อไปให้ทำการเปลี่ยน Password (ให้ทำครั้งเดียวตอน log on ครั้งแรกเท่านั้น) ดูหัวข้อ 5.3 ชั้นที่ 3

6.9.4 เอาแผ่น Poly-COM ออกจากช่องขับ A ใส่แผ่นที่มีแฟ้ม TEST.FOR และ TEST.DAT ในช่องขับ A

6.9.5 เริ่มทำการส่งแฟ้มจากแผ่นดิสเกตต์ไปเก็บ Micro VAX II โดยใช้คำสั่ง \$COPY TXA1:TEST>FOR TEST.FOR (E)

6.9.6 กด C_{Alt} เข้าไว้ แล้วกด C_{F10} ปลดปล่อยคีย์ แล้วกด C_{S} Send file:TEST.FOR (E)

ถ้าการส่งถูกต้อง คือเครื่องแสดงข้อความในแฟ้ม TEST.FOR บนจอ (คล้ายกับการใช้คำสั่ง TYPE) ไปจนจบแฟ้ม แล้วหยุดอยู่

6.9.7 กด Ctrl-Z เพื่อให้ Micro VAX II เก็บแฟ้มใน account รอจนได้ \$

6.9.8 ต่อไปส่งแฟ้ม TEST.DAT ด้วยคำสั่ง

\$COPY TXA1:TEST.DAT TEST.DAT (E)

6.9.9 กด Alt F10 และ S

Send file:TEST.DAT (E)

6.9.10 กด Ctrl-Z จะได้ \$ บนจอ

6.9.11 \$DIR (E) จะเห็น TEST.FOR;1 และ TEST.DAT;1 ใน directory

6.9.12 ตรวจสอบสิ่งที่เก็บในแฟ้ม 2 แฟ้มด้วยคำสั่ง

```
$TYPE TEST.FOR (E) และ
```

```
$TYPE TEST.DAT (E)
```

6.10 คอมไพล์และวิ่งโปรแกรมด้วยคำสั่ง

```
$FOR TEST/LIS (E) ก็ยังมีข้อผิดพลาด (compile-time error)
```

ให้ข้ามไปทำข้อ 6.13

```
$LINK TEST (E)
```

ถ้าการ link ไม่มีปัญหาเราวิ่งโปรแกรมต่อไปได้

```
$RUN TEST (E)
```

```
FORTRAN STOP
```

```
$
```

6.11 ดูใน directory โดยคำสั่ง \$DIR (E) จะเห็นแฟ้ม TEST.OUT;1 ด้วย

ตรวจดูคำตอบใน TEST.OUT โดยใช้คำสั่ง TYPE

```
$TYPE TEST.OUT (E)
```

(ใช้ปุ่ม ในการหยุดจอภาพ (Hold screen) และกด อีกเมื่อต้องการให้พิมพ์ต่อ)

6.12 ถ้าคำตอบใน TEST.OUT ถูกต้องให้สั่งพิมพ์แฟ้ม TEST.LIS และ TEST.OUT ออกที่เครื่องพิมพ์ด้วยคำสั่ง PRINT

```
$PRINT TEST.LIS,TEST.OUT (E)
```

แล้วทำการ log off จาก Micro VAX II ด้วยคำสั่ง \$LO (E)

แล้วกด เพื่อกลับมาใช้ไมโครคอมพิวเตอร์ จะได้ A>

ถ้าคำตอบใน TEST.OUT ผิด ให้ตรวจสอบว่าโปรแกรมผิดหรือข้อมูลเข้าผิด

ถ้าโปรแกรมผิดอย่างเด็ดขาด ให้ไปทำข้อ 6.13 ต่อ

ถ้าข้อมูลผิดอย่างเด็ดขาด ให้ไปทำข้อ 6.15 ต่อ

ถ้าผิดทั้งสองอย่าง ให้ไปทำข้อ 6.13 - 6.15

6.13 เมื่อมีข้อผิดพลาดใน TEST.FOR

การแก้ไขข้อผิดพลาดใน TEST.FOR ให้ใช้ Keypad editor ของ Micro VAX II ในการแก้ที่ผิด (ดูรายละเอียดจากตอนที่ 2 ของคู่มือนี้)

ก่อนแก้ไขให้ดูสิ่งที่ผิดในแฟ้ม TEST.LIS ก่อนโดยใช้คำสั่ง TYPE

\$TYPE TEST.LIS (E)


คอมพิวเตอร์จะแทรกข้อความวินิจฉัยข้อผิดพลาด (diagnostic error messages) ให้ได้คำสั่งที่ผิด เมื่อทราบที่ผิดที่จะต้องแก้แล้ว (จดใส่กระดาษไว้) ให้แก้ไขโปรแกรมใน file.FOR ในที่นี้เรียก TEST.FOR ออกมาแก้ โดยใช้คำสั่ง

\$EDIT TEST.FOR (E) หรือ

\$ED TEST.FOR (E)

editor จะแสดงบรรทัดแรกของแฟ้ม TEST.FOR ให้ บรรทัดสุดท้ายของจอจะปรากฏเครื่องหมาย * อยู่ (แสดงว่าอยู่ใน line mode)

พิมพ์ C (E) (C คือ Change ไปยัง keypad editor หรือ fullscreen editor)

เครื่องจะพิมพ์ 22 บรรทัดแรกของแฟ้มที่หน้าใหม่ของจอ (ถ้าในแฟ้มมีเกิน 22 บรรทัด) นักศึกษาใช้ปุ่ม F5, F6, F7, F8 เพื่อเลื่อนเคอร์เซอร์ขึ้น ลง ไปทางซ้าย 1 ตัวอักษร ไปทางขวา 1 ตัวอักษรตามลำดับ ใช้ปุ่ม  หรือปุ่ม Backspace เพื่อลบตัวอักษรข้างหน้าของเคอร์เซอร์ทีละตัว เมื่อได้แก้ไขข้อผิดพลาดหมดแล้ว

กด Ctrl-Z

จะได้ * บนจอ

ให้กด EXIT (E) หรือ EX (E)

เพื่อให้ editor เก็บแฟ้มที่ได้แก้ไขแล้วใน account จะเห็นข้อความบอกว่าได้เก็บแฟ้ม TEST.FOR;2 ไว้แล้ว (ถ้าไม่ต้องการเก็บที่แก้ไข ให้พิมพ์ QUIT (E) แทน)

6.14 กลับไปทำข้อ 6.10 ใหม่ จนกว่าจะได้คำตอบที่ถูกต้อง

6.15 ถ้าคำตอบผิดพลาดเนื่องมาจากข้อมูลในแฟ้ม TEST.DAT ผิด เราจะแก้ไขใน TEST.DAT ให้ทำดังนี้

\$ED 'TEST.DAT (E)

*C (E)

แก้ไขข้อมูลให้ถูกต้อง

Ctrl-Z

*EX (E) จะได้ TEST.DAT;2

6.16 ถ้าไม่ได้แก้ไขข้อผิดพลาดใน TEST.FOR นักศึกษาจึงโปรแกรมทันทีด้วยคำสั่ง

\$RUN TEST(E)

ถ้านักศึกษาต้องแก้ไขทั้งโปรแกรมใน TEST.FOR และข้อมูลใน TEST.DAT เมื่อแก้ไขแล้วต้องกลับไปทำข้อ 6.10 ใหม่ คือต้องคอมไพล์โปรแกรมใหม่ก่อนการวิ่งโปรแกรม

7. ข้อผิดพลาด (Errors)

ในการเขียนโปรแกรมให้คอมพิวเตอร์ปฏิบัติตาม เพื่อให้ได้ข้อมูลออกที่ถูกต้องตามความต้องการนั้น อาจเกิดข้อผิดพลาด 2 ชนิดใหญ่ ๆ คือ

1. Compile-time errors คือ ข้อผิดพลาดที่คอมไพเลอร์ตรวจพบและรายงานออกมาให้ทราบ เพื่อให้ทำการแก้ไขต่อไป
2. Run-time errors คือ ข้อผิดพลาดที่เกิดขึ้นเมื่อวิ่งโปรแกรม อาจทำให้คอมพิวเตอร์ไม่สามารถปฏิบัติตามจนจบตามคำสั่งได้ หรืออาจทำให้ได้คำตอบที่ไม่ถูกต้อง

เราอาจจำแนกข้อผิดพลาดข้างต้นออกเป็นชนิดย่อย ๆ ได้ดังนี้ คือ

1. Syntax errors คือ ผิดไวยากรณ์
2. Typing errors คือ พิมพ์ผิด
3. Logic errors คือ ผิดตรรกะ
4. Run-time errors คือ ผิดตอนวิ่งโปรแกรม
5. Data errors คือ ข้อมูลเข้าผิด
6. Computer hardware errors คือ เครื่องคอมพิวเตอร์ผิดหรือเสีย

7.1 Syntax and Typing errors

ข้อผิดพลาดชนิด syntax error นี้ เป็นผลมาจากการใช้กฎเกณฑ์ของภาษาอย่างไม่ถูกต้องตัวอย่างเช่น

- ไม่ใส่เครื่องหมาย comma หรือใส่นิดที่
- จำนวนวงเล็บเปิดและปิดไม่เท่ากัน เป็นต้น

การพิมพ์ที่ผิดพลาด (typing error) เป็นเหตุให้เกิด syntax error ด้วย เช่น พิมพ์จุดแทน comma การพิมพ์ชื่อตัวแปรและคีย์เวิร์ดผิดเกิดขึ้นบ่อย ๆ คอมไพเลอร์จะบอกได้ว่าคีย์เวิร์ดผิด แต่ไม่อาจบอกได้ว่าชื่อตัวแปรนั้นพิมพ์ผิด เช่นพิมพ์ NVHS แทน NVEHS เป็นต้น คอมไพเลอร์จะถือว่าเป็นตัวแปรสองตัว

7.2 Logic errors

ข้อผิดพลาดชนิดนี้เกิดขึ้นจากการออกแบบโปรแกรมไม่ดี เช่น การเปลี่ยนทิศทางตามเงื่อนไขเป็นไปอย่างไม่ถูกต้อง การปฏิบัติตามโปรแกรมย่อยไม่ถูกต้องตามขั้นตอนที่ควรจะเป็น หรือลืมเรียกใช้โปรแกรมย่อยเมื่อต้องเรียกใช้ เป็นต้น การที่จะตรวจพบข้อผิดพลาดชนิดนี้ทำได้ยาก การใช้ข้อมูลสำหรับทดสอบโปรแกรมเป็นวิธีหนึ่งที่จะตรวจพบข้อผิดพลาดชนิดนี้ การใช้ผังโปรแกรม ขั้นตอนวิธี หรือรหัสเทียม (pseudocode) จะช่วยให้ตรวจพบข้อผิดพลาดชนิดนี้ได้ด้วย

7.3 Run-time errors

ข้อผิดพลาดชนิดนี้เกิดขึ้นได้ในขณะที่วิ่งโปรแกรมเท่านั้น คอมไพเลอร์จะตรวจไม่พบข้อผิดพลาดชนิดนี้ โปรแกรมควบคุมการทำงาน (Operating system) จะเป็นผู้ตรวจพบและรายงานข้อผิดพลาดชนิดนี้ให้ทราบ ตัวอย่างเช่น การหารด้วยเลขศูนย์ การที่นิพจน์ที่จะหารากที่สอง (square root) ด้วยฟังก์ชันภายในชื่อ SQRT มีค่าเป็นลบ การใช้ครรชี่ล่าง (subscript) ของสมาชิกของอะเรย์เกินขนาดที่ระบุไว้ในคำสั่ง DIMENSION การใช้คำสั่งอ่านข้อมูลจากแฟ้มที่อ้างถึง แต่สะกดชื่อแฟ้มผิดหรือไม่มีแฟ้มดังกล่าวอยู่ เป็นต้น

7.4 Data errors

ข้อผิดพลาดชนิดนี้เกิดจากการบันทึกข้อมูลลงในแฟ้มผิดพลาด ควรทำการตรวจสอบก่อนการวิ่งโปรแกรม เช่นต้องแน่ใจว่าไม่มีข้อมูลอักขระในฟิลด์ตัวเลข (numeric field) ระวังการพิมพ์ตัวอักษรโอสลับกับตัวเลขศูนย์

7.5 Computer hardware errors

เครื่องคอมพิวเตอร์เองอาจเกิดผิดพลาดขณะปฏิบัติงาน โปรแกรมควบคุมการปฏิบัติงานที่ดีที่มีประสิทธิภาพและซับซ้อน จะรายงานข้อผิดพลาดชนิดนี้ได้ดีกว่า และสามารถทำให้โปรแกรมเดิมวิ่งได้ใหม่อีก (rerun)

7.6 ข้อผิดพลาดที่พบบ่อย ๆ ใน Source program

ข้อผิดพลาดที่พบบ่อย ๆ คือ

- ลืมกำหนดค่าเริ่มต้นให้แก่ตัวแปรที่ใช้เก็บผลบวกสะสม หรือไม่ได้กำหนดค่าให้ตัวแปรที่ใช้
- พิมพ์ชื่อผิด โดยพิมพ์ตัวอักษรสลับที่กัน เช่นพิมพ์ NEVHS แทนที่จะพิมพ์ NVEHS
- ใช้ตัวแปรผิดจุดประสงค์ เช่น เอาชื่ออะเรย์มาใช้เป็นชื่อตัวแปรทั่วไป
- พิมพ์ตัวอักษรไอและตัวเลขศูนย์สลับกัน ทั้งในตัวแปรและเลขจำนวน
- ใช้วงเล็บเปิดและปิดไม่เท่ากัน ทั้งในนิพจน์ ในคำสั่ง FORMAT และในการใช้

implied DO

- ใช้คำสั่งไม่ถูกรูปแบบของคำสั่ง
- สะกดคือเวิร์ดผิด เช่นคำว่า DIMENSION สะกดเป็น DIMENTION เป็นต้น
- ไม่กำหนดคำสั่งสุดท้ายของ loop เมื่อใช้คำสั่ง DO
- ไม่ใส่คำสั่ง ENDIF เมื่อใช้รูปแบบ IF-THEN-ELSE
- ใช้ดรรชนีล่าง (subscript) เกินค่าสูงสุดที่กำหนดไว้ในคำสั่ง DIMENSION
- เขียนนิพจน์ในภาษาคอมพิวเตอร์ไม่ถูกต้องตามนิพจน์พีชคณิตที่ต้องการ
- ในคำสั่งเช่น READ(1,10) และ WRITE(6,12) ไม่ได้กำหนดว่า 1 และ 6 หมายถึงแฟ้มใด คือลืมใส่คำสั่ง OPEN เพื่อเปิดแฟ้มที่ต้องการใช้ (ในกรณีนี้อาจรอดบเมื่อระบบถามว่า unit 1 และ unit 6 คืออะไรก็ได้)

- ระหว่าง format codes ไม่ใส่เครื่องหมาย comma
- สำหรับ literal constant ลืมใส่เครื่องหมายอะโพสโทรฟี (single quote) ปิดเปิด

- ใช้ format code ไม่ตรงกับชนิดของตัวแปร เช่น ตัวแปรแบบ integer แต่ใช้ Fw.d format code
- จำนวนค่าคงที่และจำนวนตัวแปรในคำสั่ง DATA ไม่เท่ากัน
- เมื่อต่อบรรทัดลืมนำตัวอักษร (ห้ามใช้เลขศูนย์และที่ว่าง) ในคอลัมน์ 6 ของบรรทัดที่ต่อ จะทำให้บรรทัดข้างต้นและบรรทัดที่ต่อผิดพลาด
- พิมพ์คำสั่งเกินคอลัมน์ที่ 72 ของบรรทัด (เมื่อพิมพ์คำสั่ง) ซึ่งมักจะพบในคำสั่ง FORMAT ฮาว ๆ
- มีคำสั่งอ่านข้อมูลจากแฟ้มข้อมูลเข้าจำนวนหนึ่ง แต่จำนวนเรคคอร์ดในแฟ้มนั้นมีน้อยกว่าจำนวนที่ต้องการอ่าน ในกรณีนี้ต้องตรวจสอบในแฟ้มข้อมูลเข้านั้น ๆ
- ในคำสั่ง FORMAT ของการพิมพ์ผล จำนวนตำแหน่งที่สั่งพิมพ์ใน 1 บรรทัดเกิน 132 ตำแหน่งสำหรับแคร์พิมพ์ยาวโดยทั่ว ๆ ไป หรือ 80 ตำแหน่งสำหรับกระดาษขนาด A4
- ในการใช้ intrinsic function ใช้จำนวนตัว ชนิด และขนาด (magnitude) ของตัวอาร์กิวเมนต์ (argument) ไม่ถูกต้องที่กำหนด เช่น การใช้ฟังก์ชัน SQRT นั้นอาร์กิวเมนต์ต้องมีชนิดเป็น real และมีค่ามากกว่าหรือเท่ากับศูนย์ ถ้าใส่อาร์กิวเมนต์เกิน 1 ตัว หรืออาร์กิวเมนต์มีค่าเป็นลบจะเกิดข้อผิดพลาดขึ้น
- วางคำสั่งไม่ถูกต้องตามลำดับที่ของการเรียงคำสั่งในโปรแกรม เช่น คำสั่ง DIMENSION จะต้องวางอยู่ก่อนคำสั่งปฏิบัติการใด ๆ ในโปรแกรม ดังนั้นคำสั่ง DIMENSION ต้องอยู่ก่อนคำสั่ง OPEN ถ้าคอมพิวเตอร์พบว่าคำสั่ง DIMENSION อยู่ผิดที่ นอกจากจะบอกว่ามีข้อผิดพลาดเกิดขึ้นจากการวางคำสั่งนั้นผิดแล้ว ชื่ออะเรย์ทั้งหมดในคำสั่ง DIMENSION เมื่อใช้ในโปรแกรมจะเป็นข้อผิดพลาดทุกแห่งที่ใช้

ลำดับของคำสั่งที่ FORTRAN 77 คอมไพเลอร์ส่วนมากยอมรับ คือ

PROGRAM (SUBROUTINE หรือ FUNCTION) statement
TYPE และ PARAMETER statement(s)
INTRINSIC statement(s)
DIMENSION statement(s)
COMMON statement(s)
Statement function statement(s)
Executable และ FORMAT statements
END statement