

เมื่อเราออกแบบโปรแกรมเพื่ออ่านและบันทึกข้อมูล เราสามารถเขียนโปรแกรมอ่านหรือ บันทึกแฟ้มข้อมูลคนละชุดกันทุกครั้งที่เรา run โปรแกรม ซึ่งเรียกว่า device และ file เป็นอิสระจากกัน

ระบบควบคุมการปฏิบัติงานของ VAX/VMS, device ที่เป็นอิสระจากกัน สามารถเรียกโดย **logical names** เมื่อเราเขียนโปรแกรม เราอ้างถึง input หรือ output file ตามที่ syntax ของแต่ละภาษาที่ใช้

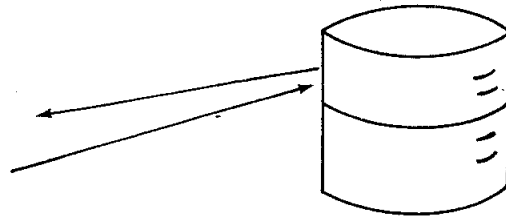
หลังจากที่โปรแกรมถูกคอมไพล์และลิงค์แต่ยังไม่ได้ run เราสามารถทำการต่อ (connection) ระหว่าง logical names ที่เราใช้ในโปรแกรมกับแฟ้มข้อมูลที่เป็นจริง (actual files) หรือ devices ที่เราต้องการใช้เมื่อ run โปรแกรม

คำสั่ง ASSIGN จะทำการต่ออันนี้ มันจะสร้างความสัมพันธ์ซึ่งกันระหว่าง logical name (หมายถึง ชื่อที่ใช้ในโปรแกรม) กับ **equivalence name** (หมายถึง แฟ้มข้อมูลที่เป็นจริง หรือ device ที่ใช้)

Terminal display

```
$ SHOW DEFAULT
DBA1 : [WELLADAY]
$ ASSIGN JANUARY.DAT INFILE
$ ASSIGN JANUARY.OUT OUTFILE
$ RUN FICA
```

Disk input/output files



DBA 1

โปรแกรม FICA.EXE มีคำสั่ง I/O เพื่อ open, read และ write ไฟล์ อ้างอิงโดย logical names INFILE และ OUTFILE

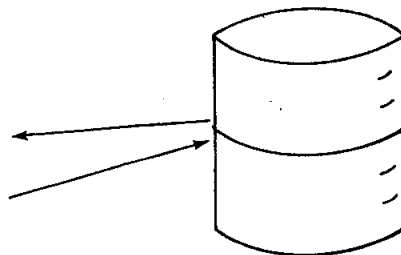
OPEN 'INFILE', 'OUTFILE'

READ INFILE

WRITE OUTFILE

SELECT A ASSIGN TO INFILE.
SELECT B ASSIGN TO OUTFILE.

```
$ ASSIGN DBA2 : FEBRUARY.DAT INFILE
$ ASSIGN DBA2 : FEBRUARY.OUT OUTFILE
$ RUN FICA
```



DBA 2

รูปแสดงการใช้ logical names

ในโปรแกรม FICA มีคำสั่ง OPEN, READ และ WRITE ในรูปทั่วไป โปรแกรมอ่านจากแฟ้มข้อมูล โดยใช้ logical name INFILE และบันทึกในแฟ้มข้อมูลโดยใช้ logical name OUTFILE

ในการ run โปรแกรมแต่ละครั้ง คำสั่ง ASSIGN จะให้ equivalence names ที่แตกต่างกัน สำหรับ INFILE และ OUTFILE

ตัวอย่างแรก โปรแกรมอ่านแฟ้มข้อมูลชื่อ JANUARY.DAT จาก device DBA1 แล้วบันทึกแฟ้มข้อมูล JANUARY.OUT บน disk device ตัวเดียวกัน

ตัวอย่างที่สอง อ่านแฟ้มข้อมูลชื่อ FEBRUARY.DAT จาก device DBA2 แล้วบันทึกแฟ้มข้อมูล FEBRUARY.OUT บน device ตัวเดียวกัน

6.1 Logical names ในคำสั่งต่าง ๆ

การใช้ logical names ไม่ได้จำกัดใช้ได้เฉพาะ application programs คำสั่งอ่าน หรือ บันทึกแฟ้มข้อมูลเช่นคำสั่ง COPY และคำสั่ง TYPE ต่างก็ต้องมี logical names สำหรับ file specification

ตัวอย่าง

```
$ ASSIGN [chuck] personnel.rec myfile
```

```
$ TYPE myfile
```

คำสั่ง ASSIGN นี้ ให้ logical name MYFILE กับแฟ้มข้อมูลชื่อ PERSONNEL.REC ซึ่งมีอยู่ในไดเรกทอรี CHUCK คำสั่ง TYPE บอก system ให้แสดงแฟ้มข้อมูลนั้นบนจอภาพของเทอร์มินัล

logical name หนึ่ง ๆ สามารถ define เฉพาะส่วนแรกของ file specification เท่านั้น

ตัวอย่าง

```
$ ASSIGN DBA2: [MALCOLM.TESTFILES] test
```

```
$ RUN test: memo
```

```
$ PRINT test: memo.lis
```

คำสั่ง ASSIGN ให้ logical name TEST กับ disk device และไดเรกทอรี DBA2: [MALCOLM.TESTFILE] ต่อจากนั้นคำสั่ง RUN execute โปรแกรม image ชื่อ MEMO.EXE และเก็บไว้ในซับไดเรกทอรีนี้ คำสั่ง PRINT พิมพ์แฟ้มข้อมูลอีกชุดหนึ่ง

ระบบควบคุมการปฏิบัติงานของเครื่องปกติจะตรวจสอบ file specification เพื่อดูว่า ส่วนของ file specification ซึ่งตามหลัง colon (:) เป็น logical name หรือไม่, ถ้าเป็น (เหมือนกับในตัวอย่างนี้) system ก็จะแทนที่ไปด้วย equivalence name

6.2 System default logical name

เมื่อเรา log in เครื่องหรือสั่ง batch job, ระบบจะจัด default logical names ให้หลายตัว ชื่อเหล่านี้จะถูกใช้โดยคำสั่ง interpreter เพื่ออ่านคำสั่งโปรแกรมของเรา แล้วพิมพ์การตอบรับ หรือข้อความบอกที่ผิด logical names เหล่านี้ได้แก่

Logical name	ใช้ (Use)
SYSS\$INPUT	default input stream ซึ่งเครื่องจะอ่านคำสั่ง และโปรแกรมของเราอ่านข้อมูล Default interactive assignment: terminal Default batch assignment: คำสั่ง procedure หรือ batch stream
SYSS\$OUTPUT	default output stream ซึ่งเครื่องจะบันทึก การตอบรับคำสั่ง และโปรแกรมของเรา บันทึกข้อมูล Default interactive assignment: terminal Default batch assignment: batch job log file
SYSS\$ERROR	default device ซึ่งเครื่องจะพิมพ์ข้อผิดพลาด ทั้งหมด และ information messages Default interactive assignment: terminal Default batch assignment: batch job log file
SYSS\$DISK	default disk device ของเรา Default assignment: set in user authorization file

เราสามารถเอา logical name เหล่านี้มาใช้ในโปรแกรม ตัวอย่างเช่น ถ้าเราเขียนโปรแกรม บันทึกเพิ่มข้อมูลไว้ใน device ชื่อ SYSS\$OUTPUT, output ไฟล์จะออกมาทางเทอร์มินัล ถ้าเรา execute โปรแกรมทันที หรือ batch job log file ถ้าเรา execute โปรแกรม ใน batch job

เราสามารถกำหนด logical name ชื่อหนึ่งให้กับ logical name อีกชื่อหนึ่งได้ ตัวอย่างเช่น การทดสอบโปรแกรม FICA ตามที่แสดงไว้ในตอนต้น เราอาจจะกำหนด logical name OUTFILE ให้กับ logical name SYSS\$OUTPUT ได้ดังนี้

```
$ ASSIGN SYSS$OUTPUT OUTFILE
```

เมื่อโปรแกรม FICA บันทึกทาง logical device OUTFILE, output จะปรากฏทางเทอร์มินัล