

บรรณานุกรม

- กลางเดือน โพนนา. ระบบสนับสนุนการตัดสินใจสำหรับการวางแผนการผลิตในโรงงานแปรรูปเนื้อไก่. สาขาวิศวกรรมอุตสาหกรรม จุฬาลงกรณ์มหาวิทยาลัย.
วิทยานิพนธ์ปริญญามหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย, 2534.
- กิตติศักดิ์ พลอยพานิชเจริญ. การวางแผนการผลิตสำหรับครอบครัว เบื้องายหินที่ทำได้ด้วยมือ. สาขาวิศวกรรมอุตสาหกรรม จุฬาลงกรณ์มหาวิทยาลัย.
วิทยานิพนธ์ปริญญามหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย, 2529.
- เจริญ สุนทรวาทิชย์. การออกแบบการผลิตและพัฒนาคงคลังสำหรับโรงงานกระดาษเหนียว. สาขาวิศวกรรมอุตสาหกรรม จุฬาลงกรณ์มหาวิทยาลัย.
วิทยานิพนธ์ปริญญามหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย, 2529.
- จิระศักดิ์ เจริญสุข. การออกแบบระบบการผลิตแบบเซลล์ในโรงงานประกอบเครื่องปรับอากาศ. สาขาวิศวกรรมอุตสาหกรรม จุฬาลงกรณ์มหาวิทยาลัย.
วิทยานิพนธ์ปริญญามหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย, 2530.
- พิภพ เล้าประจง. ระบบการควบคุมการผลิตเชิงวิศวกรรม. พิมพ์ครั้งที่ 2. กรุงเทพมหานคร: เอเชียเพรส จำกัด, 2531.
- ยีน กุสุวรรณ. แอดวานซ์ดีเบส. กรุงเทพมหานคร : เอช-เอน การพิมพ์, 2534.
- สมนึก วิสุทธิแพทย์ . การปรับปรุงแผนการผลิตของโรงงานผลิตกระป๋องโลหะขนาดเล็กในประเทศไทย. สาขาวิศวกรรมอุตสาหกรรม จุฬาลงกรณ์มหาวิทยาลัย .
วิทยานิพนธ์ปริญญามหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย , 2528.
- สมชาย พัวจินดาเนตร . การออกแบบระบบข้อเสนอแนะทางการผลิตสำหรับโรงงานผลิตเม็ดพลาสติกพีวีซี. สาขาวิศวกรรมอุตสาหกรรม จุฬาลงกรณ์มหาวิทยาลัย .
วิทยานิพนธ์ปริญญามหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย , 2528.
- ปราณี ธรรมรักษ์ และสมศักดิ์ เกรอด . เทคนิคในการออกแบบและพัฒนาระบบงานคอมพิวเตอร์. กรุงเทพมหานคร: สำนักพิมพ์ยูไนเต็ดบุคส์ , 2530.
- อัครเสน สมุทรพ่อง และพิไลวรรณ พุฒมงคล. คู่มือการใช้งาน Dbase III/III PLUS. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร: เอช-เอน การพิมพ์, 2530.

- Andrew Kusiak. Flexible manufacturing systems : Methods and studies .
Elsevier Science Publishers B.V., 1986.
- Andrew Kusiak. Intelligent manufacturing system , Prentice - Hall inc,
1986.
- Blank, J.T., Cellular Manufacturing System Reduce Setup Time, Make-
Small Lot Production Economical, Industrial Engineering,
Vol.15, No.11, PP.36-98, 1983.
- Charles S.Snead. Group technology foundation for competitive
manufacturing , Van nostrand reinhold, 1989.
- Michelle M.Vig. Dynamic rules for due-date assignment. International
journal of Production Research. Vol.29, No.7, pp.1361-1377, 1991.
- Edward Jones. Foxpro Made Easy. Osborne McGraw-Hill,1990.
- Elwood S.buffa. Production inventory system planning and control.
Richard D.irwin Inc, 1968.
- Fox Software,Inc. Foxpro : Command & Function. The Fox Holding,Inc.
, 1991.
- Garrett J.Van Ryzin. Scheduling job shop with delays. International
journal of Production Research. Vol.29, No.7, PP.1407-1422, 1991.
- Kochan, D., CAM Developments in computer-Integrated Manufacturing,
Springer-Verlag, Berlin Heidelberg, 1986.
- S. Atre.database. Structure technique for design performance and-
management. John Wiley&Sons, 1986.
- Urban Wemmerlov. Job and Family Scheduling of a Flow-Line -
Manufacturing Cell: A Simulation Study .IEE Transactions,Vol.23
,No.4, PP.383-393, 1991.

ภาคผนวก ก

รายชื่อแฟ้มข้อมูล

แฟ้มข้อมูลเปลี่ยนแปลง	รายละเอียด
Tra_po.dbf	แฟ้มข้อมูลที่ใช้รวบรวมข้อมูล เกี่ยวกับคำสั่งผลิต
Stk_rec.dbf	แฟ้มข้อมูลเกี่ยวกับการรับชิ้นงานเข้าทัศนคณคลัง
Stk_shp.dbf	แฟ้มข้อมูลเกี่ยวกับการจ่ายชิ้นงานออกจากทัศนคณคลัง

แฟ้มข้อมูลหลัก	รายละเอียด
Po.dbf	แฟ้มข้อมูลแสดงรายการคำสั่งผลิตที่ยืนยันแล้วรอการผลิต
Stk_bal.dbf	แฟ้มข้อมูลเกี่ยวกับยอดสมดุลคงเหลือของชิ้นงานใน ทัศนคณคลัง

แฟ้มข้อมูลพื้นฐาน	รายละเอียด
Part.dbf	แฟ้มข้อมูลเกี่ยวกับรายละเอียดชิ้นงาน ที่ใช้ในการผลิต เช่น วัตถุดิบที่ใช้,ขนาด,เบอร์รูปภาพ (Drawing No.)
Product.dbf	ข้อมูลเวลาในการผลิต และ เวลาตั้งเครื่องจักร เป็นต้น
Used.dbf	แฟ้มข้อมูลเกี่ยวกับรายละเอียด ของเครื่องปรับอากาศ เช่น รหัสผลิตภัณฑ์,สี,ลูกค้ำ,ชนิดของคอล์ย เป็นต้น
	แฟ้มข้อมูลเกี่ยวกับสูตรการผลิตว่า เครื่องปรับอากาศ รุ่นไหนประกอบด้วยชิ้นงานอะไรบ้าง จำนวนเท่าใด

ตารางที่ 6.1 แสดงแฟ้มฐานข้อมูลต่างๆพร้อมรายละเอียดในการเก็บข้อมูล

ภาคผนวก ข

โครงสร้างและความสัมพันธ์ของแฟ้มข้อมูล

PO.DBF

FIELD	TYPE DATA
JOB	C 10
PD-CODE	C 7
QTY	N 4
DATE	D 8
REMARK	C 25

USED.DBF

FIELD	TYPE DATA
CODE	C 7
PD-CODE	C 1
USED	N 4
OPTION	C 2
PRIORITY	N 2

PRODUCT.DBF

FIELD	TYPE DATA
PD-CODE	C 7
CUSTOMER	C 15
UE	C 5
CM	C 5
COLOR	C 5
APPLICAT	C 15
FIN	C 3
VAL	C 3
COMPRESS	C 14
HEATE	C 5
SYSTEM	C 15
PACKING	C 15
KIT	C 3
HILO	C 10
TRANSFER	C 3
SOCKET	C 10
OPTION	C 2
REMARK	C 10

PART.DBF

FIELD	TYPE DATA
CODE	C 7
NAME	C 40
MATERIAL	C 5
SIZE	C 10
DRAWING	C 8
TURRET	C 3
LVD	C 7
AMADA	C 7
C_LVD	N 4
C_10	N 4
C_4	N 4
P_SMA	N 4
P_MED	N 4
P_LAR	N 4
P_TUR	N 4
B_LVD	N 4
B_AMA	N 4
B_CHA	N 4
B_KLE	N 4
ARC	N 4
SOLDER	N 4
FACE	N 4
SW	N 4

REMARK : TYPE C = CHARACTER
 N = NUMERIC
 D = DATE

ตารางที่ 6.2 แสดงความสัมพันธ์ระหว่างแฟ้มข้อมูลหลัก กับแฟ้มข้อมูลภายนอก

STK_REC.DBF

FIELD	TYPE
CODE	C 7
DATE	D 8
QTY	N 4

STK_SHP.DBF

FIELD	TYPE
CODE	C 7
DATE	D 8
QTY	N 4

STK_BAL.DBF

FIELD	TYPE
CODE	C 7
DATE	D 8
QTY	N 4

TRA_PO.DBF

FIELD	TYPE
JOB	C 10
PD-CODE	C 7
QTY	N 4
DATE	D 8
REMARK	C 25

PO.DBF

FIELD	TYPE
JOB	C 10
PD-CODE	C 7
QTY	N 4
DATE	D 8
REMARK	C 25

REMARK: TYPE C = CHARACTER
 N = NUMERIC
 D = DATE

ตารางที่ 6.3 แสดงความสัมพันธ์ระหว่างแฟ้มข้อมูลหลัก กับแฟ้มข้อมูลเปลี่ยนแปลง

ภาคผนวก ค

ตัวอย่างการหาเวลาในการผลิตชิ้นงานโลหะแผ่น

TIME STUDY ON FABRICATE LINE

MACHINE : LVD (SHEAR MACHINE)
 MAN - USED : 1 MAN/PART
 PART : FRONT COVER MFC (600,900)

OPERATION	UNIT PER LOAD	TIME (MIN)	REMARK
1) LOAD PART TO WORK ATTACH BACK- GUAGE	1	0.080	
2) PRESS STAND TO START SHEAR + SHEAR SHEET METAL	1	0.093	
3)UNLOAD REMAIN PART	1	0.070	
TOTAL		0.243	

$$\text{UNIT PER HOUR (UPH)} = (\text{UNIT PER LOAD} * 60) / (\text{TOTAL TIME} * 1.1)$$

$$= 224 \text{ PCS/HOURS}$$

REMARK : 20 RAW DATA FROM TIME STUDY AVERAGE TO TIME (MIN)
 :NOT INCLUDE SETUP TIME AND MATRIAL HANDLING TIME

TIME STUDY ON FABRICATE LINE

MACHINE : TURRET (PUNCH MACHINE)
 MAN - USED : 1 MAN/PART
 PART : FRONT COVER MFC (600,900)

OPERATION	UNIT PER LOAD	TIME (MIN)	REMARK
1) LOAD PART TO WORK ATTACH SETUP GUAGE	1	0.08	
2) PRESS STAND TO START PUNCH + PUNCH SHEET METAL	1	0.67	
4)UNLOAD PART	1	0.08	
TOTAL		0.83	

$$\text{UNIT PER HOUR (UPH)} = (\text{UNIT PER LOAD} * 60) / (\text{TOTAL TIME} * 1.1)$$

$$= 65 \text{ PCS/HOURS}$$

REMARK : 20 RAW DATA FROM TIME STUDY AVERAGE TO TIME (MIN)
 :NOT INCLUDE SETUP TIME AND MATRIAL HANDLING TIME

TIME STUDY ON FABRICATE LINE

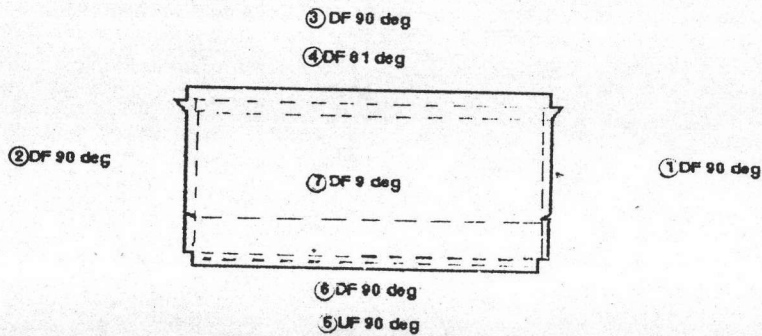
MACHINE : LVD (PRESS BAKE MACHINE)
 MAN - USED : 2 MAN/PART
 PART : FRONT COVER MFC (600,900)

OPERATION	UNIT PER LOAD	TIME (MIN)	REMARK
1) LOAD PART TO WORK ADJUST PART BEFORE RUN	1	0.060	
2) PRESS NO.1	1	0.073	
3) FLIP PART + PRESS NO.2	1	0.130	
4) FLIP PART + PRESS NO.3,4	1	0.200	
5) FLIP PART + PRESS NO.5	1	0.123	
6) FLIP PART + PRESS NO.6	1	0.183	
7) PRESS NO.7	1	0.103	
8) UNLOAD PART ON SHELF	1	0.083	
TOTAL		0.955	

$$\text{UNIT PER HOUR (UPH)} = (\text{UNIT PER LOAD} * 60) / (\text{TOTAL TIME} * 1.1)$$

$$= 57 \text{ PCS/HOURS}$$

REMARK : 20 RAW DATA FROM TIME STUDY AVERAGE TO TIME (MIN)
 :NOT INCLUDE SETUP TIME AND MATERIAL HANDLING TIME

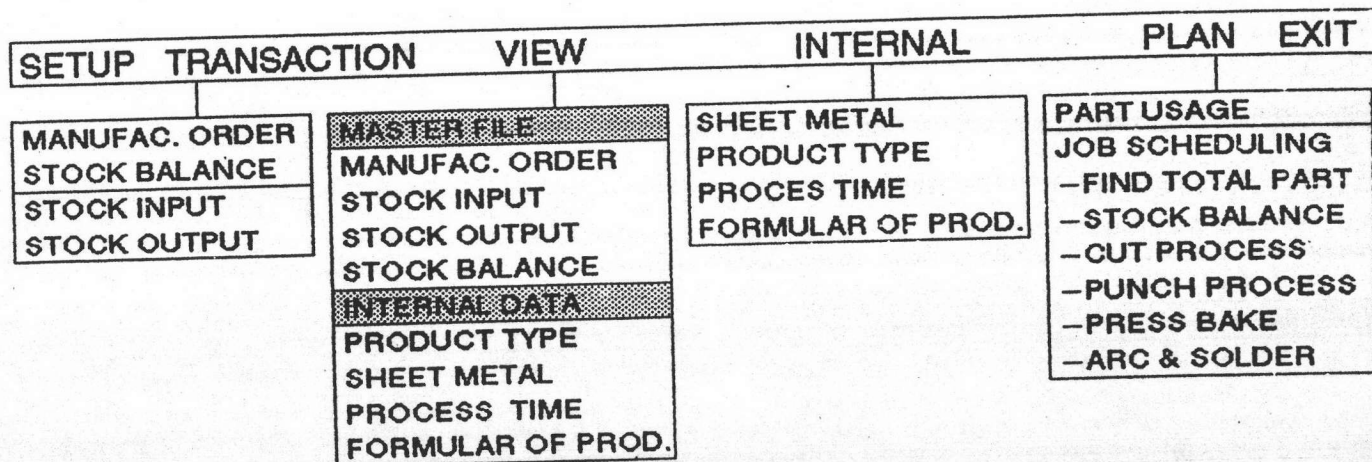


ภาคผนวก ง

ตัวอย่างการใช้งานโปรแกรมการจัดลำดับงาน

1. การเข้าคอมพิวเตอร์เบื้องต้น

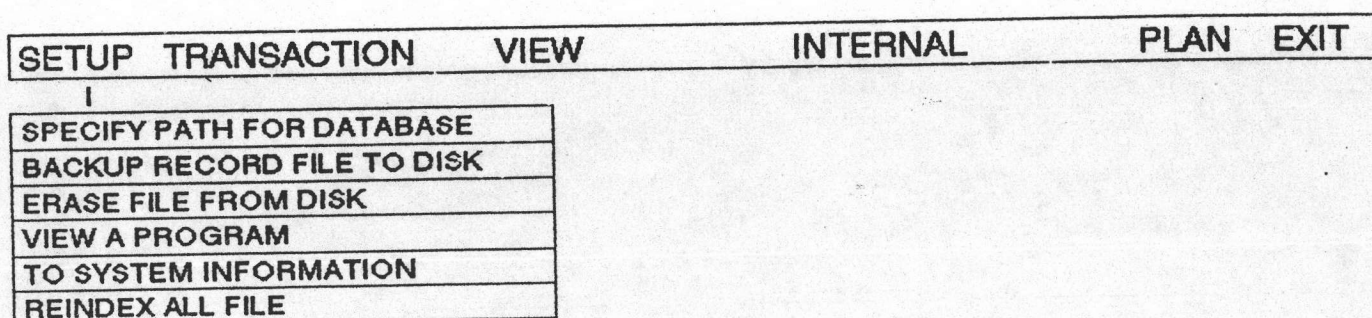
- ก. วิธีที่ 1 เมื่อได้เปิดเครื่องคอมพิวเตอร์ ให้เข้าร่ายย่อยที่มีโปรแกรม FoxPro V.2 อยู่ จากนั้นพิมพ์เรียก FOX UNIFAB เช่น ถ้าอยู่ร่าย Fox ก็เรียก C:\FOX\FOX UNIFAB เป็นต้น
- ข. วิธีที่ 2 เมื่อเปิดเครื่องคอมพิวเตอร์ ให้เข้าร่ายย่อยที่มีโปรแกรม FoxPro อยู่ จากนั้นเรียกโปรแกรม FOXPRO V.2 โดยการพิมพ์ Foxpro เช่น C:\FOX\FOXPRO จากนั้นก็กด CTRL +D แล้วเลื่อน ลูกศรขึ้นและลง ไปที่โปรแกรม UNIFAB และกด ENTER ก็จะสามารถเข้าสู่โปรแกรมการจัดลำดับงานได้ ดังรูปที่ 6.1



รูปที่ 6.2 แสดงหน้าจอรายการหลักทั้งหมด (Main Menu) ในการทำงานกับโปรแกรม

2. การใช้โปรแกรมการจัดลำดับงาน

- ก. หมวด SETUP จะมีรายการย่อย 6 รายการให้เลือกโดย ลูกศรขึ้นลง ดังรูปที่ 6.2 ข้างล่าง



รูปที่ 6.2 แสดงหน้าจอในการทำงาน หมวด SETUP ของโปรแกรม

1. SPECIFY PATH FOR DATABASE

ใช้กำหนดร่ายย่อยในการใช้งานโปรแกรม เมื่อเลือกร่ายย่อยเรียบร้อยแล้ว จะออกจากรานการนี้กด ESC

2. BACKUP RECORD FILE TO DISK

ใช้ในการทำสำเนาข้อมูลที่ต้องการลงแผ่น DISKETTE เก็บไว้นอกเครื่องคอมพิวเตอร์โดยโปรแกรมจะถามชื่อไฟล์ และ DRIVE ที่ต้องการนำไปเก็บ เมื่อเลือกทำสำเนาเรียบร้อยแล้วจะออก กด ESC

3. ERASE FILE FROM DISK

ใช้ลบไฟล์ที่ไม่ต้องการออกจากเครื่องคอมพิวเตอร์ ใส่ชื่อไฟล์ที่ต้องการจะลบลงโปรแกรมจะถามเพื่อความแน่ใจ ในการลบ ถ้าแน่ใจกด Y โปรแกรมจะทำการลบไฟล์ชื่อนั้นออกไป หากไม่แน่ใจกด N โปรแกรมจะยังไม่ทำอะไรและออกมา เมื่อเลือกเสร็จจะออก กด ESC

4. VIEW A PROGRAM

ใช้สำหรับ โปรแกรมเมอร์ ในการเข้าไปตรวจสอบโปรแกรมที่เขียนไว้ แต่มีให้แก้ไขโปรแกรม (หากแก้ไขโปรแกรมจะ เกิดความผิดพลาดได้) เมื่อเลือกเสร็จจะออกกด ESC

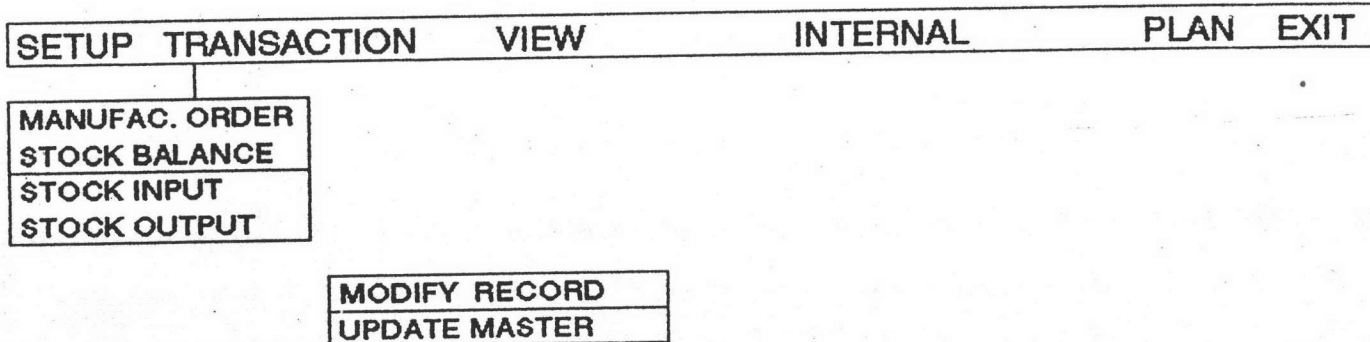
5. TO SYSTEM INFORMATION

ใช้สำหรับ ดูระบบความจำ และ เครื่องมืออุปกรณ์ในเครื่องคอมพิวเตอร์ที่ผู้ใช้งานกำลังทำงานอยู่ เมื่อเลือกเสร็จจะออก กด ESC

6. REINDEX ALL FILE

ใช้สำหรับ จัดเรียงข้อมูลใหม่ทุกไฟล์ (ปกติไม่ต้องใช้ ยกเว้นมีการเปลี่ยนแปลงข้อมูลในข้อมูลภายใน INTERNAL FILE จึงอาจจะมีการเรียงลำดับใหม่) เมื่อเลือกเสร็จ และทำงานเรียบร้อยแล้ว จะออก กด ESC

ข. หมวด TRANSACTION ใช้สำหรับการป้อน แก้ไข หรือ ลบ ข้อมูล ที่มีการเปลี่ยนแปลงบ่อย ๆ จะมีรายการย่อยอยู่ 4 รายการให้เลือก ดังรูปที่ 6.3 โดยเลื่อนลูกศร ขึ้นลง เมื่อต้องการเลือก ให้กด ENTER ต่อจากนั้นโปรแกรมจะให้เลือก MODIFY RECORD หรือ UPDATE MASTER



รูปที่ 6.3 แสดงหน้าจอในการทำงาน หมวด TRANSACTION ของโปรแกรม

1. MANUFACTURING ORDER เมื่อเลือก ตัวเลือกนี้ จะมีรายการย่อย MODIFY RECORD กับ UPDATE MASTER ซึ่งหากเลือก MODIFY RECORD โปรแกรมก็จะเข้าไปการแก้ไข เพิ่มเติมข้อมูลป้อนงาน ที่จะทำการวางแผนการจัดลำดับงาน โดยสามารถป้อนได้ไม่เกิน 8 งาน (JOB ORDER) โดยข้อมูลที่ต้องการมีอยู่ 4 ประเภทข้อมูลเท่านั้น

- เลขที่งาน (JOB NO.) คือ เลขที่ ทำชิ้นการผลิตภัณฑ์
- รหัสผลิตภัณฑ์ (PRODUCT CODE) คือ รหัสผลิตภัณฑ์ที่ทางโรงงานตั้งไว้
- วัน เดือน ปี ที่ต้องการ (DATE REQUIREMENT) คือ วันที่ ที่ตั้งไว้ว่าจะทำการผลิต ผลิตภัณฑ์นี้
- จำนวน (QTY) คือ จำนวนเครื่องที่ต้องการทำการผลิต

จากนั้นโปรแกรมจะ ไปทำการค้นหาข้อมูลอื่นๆ ออกมาแสดงลงในตัวอย่างหน้าจอ ดังรูปที่ 6.4

MANUFACTURING ORDER					
JOB NO :	JB001	PRODUCT CODE	: 7402J01	CUSTOMER :	UNI-AIRE
COLOR :	U-23	DATE REQUIRE	: 03/05/92	QTY	: 100.00
UE	: MFC 1600	THERMO :	"OTM"		
CUSTOMER MODEL:	FUB	TRANSFER :	YES		
APPLICATION	: COOL ONLY	SOCKET :	NO (C3AFIM)		
VAL	: H/U				
ELECTRIC SYSTEM :	24 V,50 &60 HZ				
REMARK	:				

FIND	NEXT	BEFOR	TOP	LAST	EDIT	ADD	DELETE	PRINT	QUIT
------	------	-------	-----	------	------	-----	--------	-------	------

รูปที่ 6.4 แสดงหน้าจอในการทำงานการป้อนคำสั่งผลิตของโปรแกรม

ส่วนเมนูย่อยที่บรรทัดล่างสุดจะทำงานดังนี้

- FIND เมื่อกดเลือกรายการนี้ โปรแกรมจะขึ้นหน้าต่าง JOB NO. ให้ผู้ใช้เลือก
- NEXT เมื่อกดเลือกรายการนี้ โปรแกรมจะเลื่อน JOB NO. ไป 1 รายการ
- BEFOR เมื่อกดเลือกรายการนี้ โปรแกรมจะถอยหลัง 1 JOB NO.
- TOP เมื่อกดเลือกรายการนี้ โปรแกรมจะขึ้นไปบนสุดของไฟล์ข้อมูล (รายการที่ 1)
- LAST เมื่อกดเลือกรายการนี้ โปรแกรมจะลงไปที่ท้ายสุดของไฟล์ข้อมูล (รายการสุดท้าย)
- EDIT เมื่อกดเลือกรายการนี้ โปรแกรมจะเข้าไปแก้ไขข้อมูลในรายการขณะนั้น
- ADD เมื่อกดเลือกรายการนี้ โปรแกรมจะเพิ่มรายการข้อมูลใหม่
- DELETE เมื่อกดเลือกรายการนี้ โปรแกรมจะลบข้อมูลในรายการนั้นออก
- PRINT เมื่อกดเลือกรายการนี้ โปรแกรมจะพิมพ์ข้อมูลในไฟล์ข้อมูลนี้ออกมา
- QUIT เมื่อกดเลือกรายการนี้ โปรแกรมจะออกจากเมนูหัวข้อนี้

และหากออกจากรายการนี้ ก็จะพบรายการ UPDATE MASTER หากกดเลือกหัวข้อนี้โปรแกรม จะทำการ ปรับทันกาล (UPDATE) ข้อมูลลงใน MASTER FILE เพื่อไว้ใช้ในการคำนวณวางแผนต่อไปแต่หากไม่ต้องการ UPDATE ข้อมูล ก็กด ESC ออกไป

2. STOCK INPUT ใช้สำหรับการป้อน ชิ้นงานที่มีอยู่ในพัสดุคงคลัง หรือที่ผลิตเหลือมาใช้ เก็บเข้าพัสดุคงคลัง โดยข้อมูลที่ต้องป้อนมี รหัสชิ้นงาน ,วันที่ และ จำนวนในการนำเข้าดังรูปที่ 6.5

PART INPUT HEADER												
CODE : 6205P02												
ACM MODEL	10	11	12	15	16	18	20	25	30	36	50	60
USED PER SET	1	1										

STOCK INPUT		
CODE	DATE	QTY
6205P02	01/06/92	10
6205P02	02/06/92	20
TOTAL		30

STOCK EDIT , ADD , DEL		
CODE	DATE	QTY
6205P02	01/06/92	10
6205P02	02/06/92	20

FIND	NEXT	BEFOR	EDIT	ADD	DELETE	QUIT
------	------	-------	------	-----	--------	------

รูปที่ 6.5 แสดงหน้าจอในการทำงานการใส่ชิ้นงานเข้า STOCK ของโปรแกรม

ส่วนเมนูย่อยที่บรรทัดล่างสุดจะทำงานดังนี้

- FIND เมื่อกดเลือกรายการนี้ โปรแกรมจะขึ้นหน้าต่าง PART NAME ให้ผู้ใช้เลือก
- NEXT เมื่อกดเลือกรายการนี้ โปรแกรมจะเลื่อน PART NAME ไป 1 รายการ
- BEFOR เมื่อกดเลือกรายการนี้ โปรแกรมจะถอยหลัง 1 PART NAME
- EDIT เมื่อกดเลือกรายการนี้ โปรแกรมจะเข้าไปแก้ไขข้อมูลในรายการขณะนั้น
- ADD เมื่อกดเลือกรายการนี้ โปรแกรมจะเพิ่มรายการข้อมูลใหม่
- DELETE เมื่อกดเลือกรายการนี้ โปรแกรมจะลบข้อมูลในรายการนั้นออก
- QUIT เมื่อกดเลือกรายการนี้ โปรแกรมจะออกจากเมนูหัวข้อนี้

และหากออกจากรายการนี้ ก็จะพบรายการ UPDATE MASTER หากกดเลือกหัวข้อนี้โปรแกรม จะทำการ ปรับทันกาล (UPDATE) ข้อมูลลงใน MASTER FILE ของข้อมูล ยอดชิ้นงานเข้า เพื่อไว้ใช้ในการคำนวณยอดสมดุล (STOCK BALANCE)ต่อไป

3. STOCK OUTPUT ใช้สำหรับการป้อน ชิ้นงานที่ถูกเบิกออกจากพัสดุคงคลัง หรือต้องการ ทำให้ยอดชิ้นงานในพัสดุคงคลังน้อยลง โดยข้อมูลที่ต้องการมี รหัสชิ้นงาน , วันที่เบิกออกและ จำนวนที่เบิกออก

PART OUTPUT HEADER										
CODE : 6205P02										
ACM MODEL	10	11	12	15	16	18	20	25	30	36 50 60
USED PER SET	1	1								

STOCK INPUT		
CODE	DATE	QTY
6205P02	01/06/92	10
6205P02	02/06/92	10
TOTAL		20

STOCK EDIT , ADD , DEL		
CODE	DATE	QTY
6205P02	01/06/92	10
6205P02	02/06/92	10

FIND	NEXT	BEFOR	EDIT	ADD	DELETE	QUIT
------	------	-------	------	-----	--------	------

รูปที่ 6.6 แสดงหน้าจอในการทำงานการนำชิ้นงานออกจาก STOCK ของโปรแกรม

ส่วนเมนูย่อยที่บรรทัดล่างสุดจะทำงานดังนี้

- FIND เมื่อกดเลือกรายการนี้ โปรแกรมจะขึ้นหน้าต่าง PART NAME ให้ผู้ใช้เลือก
- NEXT เมื่อกดเลือกรายการนี้ โปรแกรมจะเลื่อน PART NAME ไป 1 รายการ
- BEFOR เมื่อกดเลือกรายการนี้ โปรแกรมจะถอยหลัง 1 PART NAME
- EDIT เมื่อกดเลือกรายการนี้ โปรแกรมจะเข้าไปแก้ไขข้อมูลในรายการขณะนั้น
- ADD เมื่อกดเลือกรายการนี้ โปรแกรมจะเพิ่มรายการข้อมูลใหม่
- DELETE เมื่อกดเลือกรายการนี้ โปรแกรมจะลบข้อมูลในรายการนั้นออก
- QUIT เมื่อกดเลือกรายการนี้ โปรแกรมจะออกจากเมนูหัวข้อนี้

และหากออกจากรายการนี้ ก็จะพบรายการ UPDATE MASTER หากกดเลือกหัวข้อนี้โปรแกรม จะทำการ ปรับทันกาล (UPDATE) ข้อมูลลงใน MASTER FILE ของยอดจำนวนชิ้นงานออก เพื่อไว้ใช้ในการคำนวณยอดสมดุล (STOCK BALANCE) ต่อไป หากไม่ต้องการเลือกก็กด ESC ออกไป

4. STOCK BALANCE ใช้สำหรับการดูจำนวนชิ้นงานที่สมดุลยอด BALANCE ชิ้นงาน เรียบร้อยแล้ว ดังรูปที่ 6.7

STATUS STOCK BALANCE			
CODE	NAME	TYPE	CURRENT TIME
6205P02	BOTTOM PLATE (10 -11)	ACM	02/06/92

INPUT			OUTPUT		
CODE	DATE	QTY	CODE	DATE	QTY
6205P02	01/06/92	10			
6205P02	02/06/92	10			

TOTAL =	20	TOTAL = 0	BALANCE = 20
FIND	NEXT	BEFOR	QUIT

รูปที่ 6.7 แสดงหน้าจอในการทำงานการนำชิ้นงานออกจาก STOCK ของโปรแกรม

- FIND เมื่อกดเลือกรายการนี้ โปรแกรมจะขึ้นหน้าต่าง PART NAME ให้ผู้ใช้เลือก
- NEXT เมื่อกดเลือกรายการนี้ โปรแกรมจะเลื่อน PART NAME ไป 1 รายการ
- BEFOR เมื่อกดเลือกรายการนี้ โปรแกรมจะถอยหลัง 1 PART NAME
- QUIT เมื่อกดเลือกรายการนี้ โปรแกรมจะออกจากเมนูหัวข้อนี้

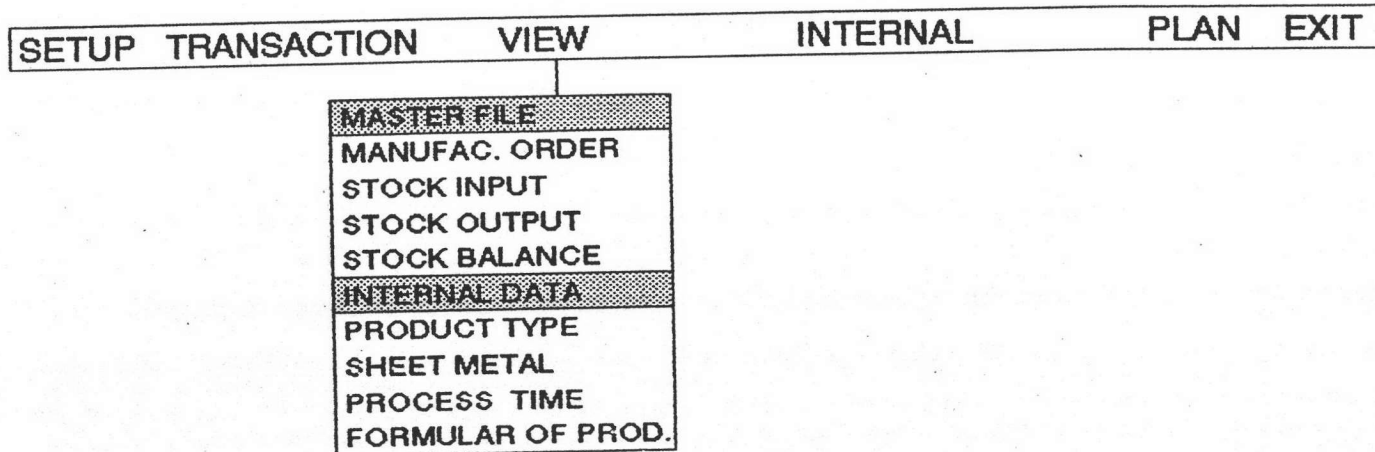
และหากออกจากรายการนี้ ก็จะพบรายการ UPDATE MASTER หากกดเลือกหัวข้อนี้โปรแกรม จะทำการ ปรับทันกาล (UPDATE) ข้อมูลลงใน MASTER FILE ของยอดสมดุลชิ้นงาน (BALANCE STOCK) เพื่อไว้ใช้ในการคำนวณวางแผนการผลิตต่อไป หากไม่ต้องการ UPDATE ข้อมูล ก็กด ESC ออกไป

ค. หมวด VIEW ใช้สำหรับการตรวจสอบข้อมูลทั้งหมด ก่อนทำการวางแผนการจัดลำดับงาน ว่าถูกต้องหรือไม่ มี 8 รายการที่ต้องการ โดยจะมี MASTER FILE อยู่ 4 ฐานข้อมูล และมี INTERNAL DATA อยู่ 4 ฐานข้อมูล ซึ่งหากทำการตรวจสอบหมดเรียบร้อยแล้วจะออก กด ESC โดยมีรายละเอียดข้อมูลที่ใช้ในการตรวจสอบ 8 รายการดังนี้

- | | | |
|------------------------|----------------------------------|-------------------|
| 1. MANUFACTURING ORDER | ใช้พิจารณา ดูใบสั่งงาน | ว่าถูกต้องหรือไม่ |
| 2. STOCK INPUT | ใช้พิจารณา ดูยอดชิ้นงานเข้า | ว่าถูกต้องหรือไม่ |
| 3. STOCK OUTPUT | ใช้พิจารณา ดูยอดชิ้นงานออก | ว่าถูกต้องหรือไม่ |
| 4. STOCK BALANCE | ใช้พิจารณา ดูยอดสมดุลชิ้นงาน | ว่าถูกต้องหรือไม่ |
| 5. PRODUCT TYPE | ใช้พิจารณา ดูรายละเอียดผลิตภัณฑ์ | ว่าถูกต้องหรือไม่ |

- | | | | |
|---------------------------|------------|-------------------|-------------------|
| 6. SHEET METAL | ใช้พิจารณา | รายละเอียดชิ้นงาน | ว่าถูกต้องหรือไม่ |
| 7. PROCESS TIME | ใช้พิจารณา | เวลาการผลิต | ว่าถูกต้องหรือไม่ |
| 8. FORMULAR OF PRODUCTION | ใช้พิจารณา | สูตรการผลิต | ว่าถูกต้องหรือไม่ |

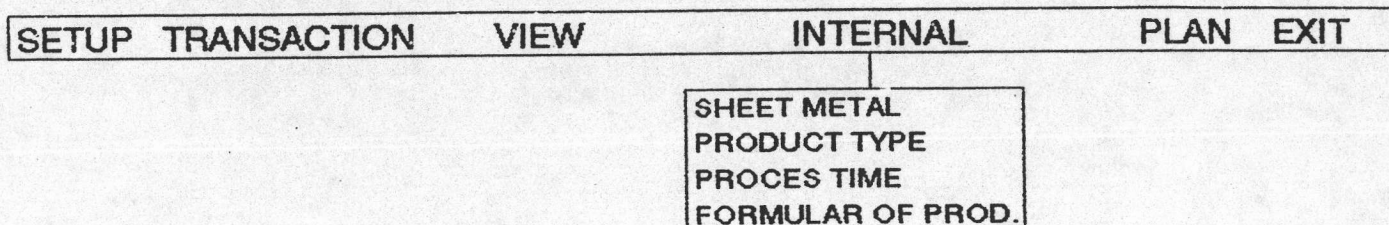
ซึ่งงานการทำงานโปรแกรมสามารถแสดงดังรูปที่ 6.8



รูปที่ 6.8 แสดงหน้าจอในการใช้งานรายการ VIEW ของโปรแกรม

โดยทุกรายการ VIEW หากต้องการ เพิ่มข้อมูลกด CTRL + N แล้วเพิ่มข้อมูลได้ทันที แต่ถ้าต้องการลบข้อมูลในรายการใด (RECORD) ก็กด CTRL + T และเมื่อจะออก ให้กด CTRL + W ก็จะได้เก็บข้อมูล (Update Data) ใหม่นี้เข้าไป แต่หากกด ESC โปรแกรมออกไปโดยไม่ได้ UPDATE ข้อมูล

ง. หมวด INTERNAL ใช้สำหรับการป้อน แก้ไข และ ลบ ข้อมูล ที่มีการเปลี่ยนแปลงน้อยก่อนทำการวางแผนการจัดลำดับงาน โดยมีเมนูคำสั่งจอภาพช่วยในการทำงานมีทางเลือกอยู่ 4 หมวด โดยลูกศร ขึ้นลงในการเลือก เมื่อเลือกเสร็จจะออก กด ESC



รูปที่ 6.9 แสดงหน้าจอในการใช้งานหมวด INTERNAL ของโปรแกรม

1. หมวด SHEET METAL ใช้ป้อน แก๊ซ หรือลอบ รายละเอียด ข้อมูลชิ้นส่วน
 ดังแสดงดังรูปที่ 6.10

PART SPECIFICATION				
CODE	: 6205P02	NAME	: BOTTOM PLATE (10-12)	
MATERIAL	: EG # 19	MODEL	USE/SET	PRIORITY
SIZE (mm *mm)	: 220/1/50	10	1	2
		11	1	
		12		
DRAWING	: U-23	15		
		16		
TURRET PROG.	: 252	18		
		20		
LVD PRG	:	25		
		30		
AMADA PRG	:	36		
		50		
		60		
FIND NEXT BEFOR TOP LAST EDIT ADD DELETE PRINT QUIT				

รูปที่ 6.10 แสดงหน้าจอในการใช้งานการป้อนรายละเอียดข้อมูลชิ้นงานของโปรแกรม

- FIND เมื่อกดเลือกรายการนี้ โปรแกรมจะขึ้นหน้าต่าง PART NAME ให้ผู้ใช้เลือก
- NEXT เมื่อกดเลือกรายการนี้ โปรแกรมจะเลื่อน PART NAME ไป 1 รายการ
- BEFOR เมื่อกดเลือกรายการนี้ โปรแกรมจะถอยหลัง 1 PART
- EDIT เมื่อกดเลือกรายการนี้ โปรแกรมจะเข้าไปแก้ไขข้อมูลในรายการขณะนั้น
- ADD เมื่อกดเลือกรายการนี้ โปรแกรมจะเพิ่มรายการข้อมูลใหม่
- DELETE เมื่อกดเลือกรายการนี้ โปรแกรมจะลบข้อมูลในรายการนั้นออก
- QUIT เมื่อกดเลือกรายการนี้ โปรแกรมจะออกจากเมนูหัวข้อนี้

2. หมวด TYPE OF PRODUCT ใช้ป้อน แก๊ซ หรือลอบ รายละเอียด ข้อมูลผลิตภัณฑ์
 ดังรูปที่ 6.11

AIR PRODUCT SPECIFICATION			
PRODUCT CODE	: 7205E01	CUSTOMER UNI - AIRE UE	: ACM CM : ACM
VAL	: B/B	FIN COIL	: AL HI & LO : NO
ELECTRIC SYS	: 220/1/50	COMPRESSOR : TECUMSHE	
COLOR	: U-23	HEATE	: NO TIMER : NO
PACKING	: EXPORT + FOAM		
FIND NEXT BEFOR TOP LAST EDIT ADD DELETE PRINT QUIT			

รูปที่ 6.11 แสดงหน้าจอในการใช้งานการป้อนรายละเอียดข้อมูลชนิดของแอร์ของโปรแกรม

- FIND เมื่อกดเลือกรายการนี้ โปรแกรมจะขึ้นหน้าต่าง JOB NO. ให้ผู้ใช้เลือก
- NEXT เมื่อกดเลือกรายการนี้ โปรแกรมจะเลื่อน JOB NO. ไป 1 รายการ
- BEFOR เมื่อกดเลือกรายการนี้ โปรแกรมจะถอยหลัง 1 JOB NO.
- EDIT เมื่อกดเลือกรายการนี้ โปรแกรมจะเข้าไปแก้ไขข้อมูลในรายการขณะนั้น
- ADD เมื่อกดเลือกรายการนี้ โปรแกรมจะเพิ่มรายการข้อมูลใหม่
- DELETE เมื่อกดเลือกรายการนี้ โปรแกรมจะลบข้อมูลในรายการนั้นออก
- QUIT เมื่อกดเลือกรายการนี้ โปรแกรมจะออกจากเมนูหัวข้อนี้

3. หมวด PROCESS TIME ใช้ป้อน แก๊ส หรือลบบ รายละเอียด ข้อมูลเวลาการผลิตชิ้นงาน และเวลาในการตั้งเครื่องจักร ของทุกกระบวนการในการผลิต ตั้งแต่ เครื่องจักรตัด , เครื่องจักรปั๊ม , เครื่องจักรพับ , จุดอาร์คและตกแต่ง ดังรูปที่ 6.12

FABRICATE PROCESS		CODE : 6205P02		BOTTOM PLATE (10-11)	
SHEAR TIME MAN			PRESS BAKE MAN		
LVD :	0.30	2	LVD :	0.00	2
S023 :	0.30		AMADA:	0.40	
S024 :	0.00		KLEEN:	0.00	
			CHAI :	0.00	
PUNCH TIME			24 "	:0.45	
TURRET :	0.60		ARC & SOLDER TIME		
REXROT :	0.00		SPOT W :	2.00	
UNIPUN :	0.00		SUR FI. :	2.50	
SMALL :	0.00		ARC :	0.40	
(1,2,3,4)			SOLDER :	5.50	
MEDIUM :	0.24				
(5,6)					
LARGE :	0.00				
(7,8,9,10,11,12,13)					
SETUP TIME (MIN)					
SHEAR GROUP					
LVD : 5.00					
S023 : 7.00					
S024 : 0.00					
PUNCH GROUP					
TURRET :15.00 SMA :0.00					
REXROT :0.00 MED :10.00					
UNIPUN :0.00 LAR :0.00					
PRESS BAKE GROUP					
LVD :0.00					
AMADA :15.00					
KLEEN :0.00					
CHAI :0.00					
24 " :15.00					
FIND NEXT BEFOR TOP LAST EDIT ADD DELETE PRINT QUIT					

รูปที่ 6.12 แสดงหน้าจอในการใช้งานการป้อนรายละเอียดข้อมูลเวลาการผลิตชิ้นงานของโปรแกรม

- FIND เมื่อกดเลือกรายการนี้ โปรแกรมจะขึ้นหน้าต่าง PART NAME ให้ผู้ใช้เลือก
- NEXT เมื่อกดเลือกรายการนี้ โปรแกรมจะเลื่อน PART NAME ไป 1 รายการ
- BEFOR เมื่อกดเลือกรายการนี้ โปรแกรมจะถอยหลัง 1 PART
- EDIT เมื่อกดเลือกรายการนี้ โปรแกรมจะเข้าไปแก้ไขข้อมูลในรายการขณะนั้น
- ADD เมื่อกดเลือกรายการนี้ โปรแกรมจะเพิ่มรายการข้อมูลใหม่
- DELETE เมื่อกดเลือกรายการนี้ โปรแกรมจะลบข้อมูลในรายการนั้นออก
- QUIT เมื่อกดเลือกรายการนี้ โปรแกรมจะออกจากเมนูหัวข้อนี้

4. หมวด FORMULAR OF PRODUCTION ใช้ป้อน แก๊ซ หรือลบ รายละเอียด ข้อมูลเกี่ยวกับสูตรการผลิตว่าแอร์รุ่นใดใช้ ชิ้นงานโลหะแผ่นอะไรบ้างและจำนวนที่ใช้เป็นเท่าใด ดังแสดงในรูปที่ 6.13

CODE : 6205P02		NAME : BOTTOM PLATE (10-12)	
MODEL	USE./SET	OPTION	OPTION MEAN
10;A	1	Z	N = NORMAL PART
11;B	1	Z	C = CONTROL BOX 'C' PART
12;C			H = HEATER PART
15;D			CH = CONTROL + HEATER PART
16;E			B = BACK PANEL BV
18;F			P = BACK PANEL PV
20;G			O = BACK PANEL OS
25;H			S = BACK PANEL SS
30;L			Z = CHOICE PART
36;J			R = ROTARY PART
50;N			
60;O			

FIND	NEXT	BEFOR	TOP	LAST	EDIT	ADD	DELETE	PRINT	QUIT
------	------	-------	-----	------	------	-----	--------	-------	------

รูปที่ 6.13 แสดงหน้าจอในการใช้งานการป้อนรายละเอียดข้อมูลสูตรการผลิตของโปรแกรม

- FIND เมื่อกดเลือกรายการนี้ โปรแกรมจะขึ้นหน้าต่าง PART NAME ให้ผู้ใช้เลือก
- NEXT เมื่อกดเลือกรายการนี้ โปรแกรมจะเลื่อน PART NAME ไป 1 รายการ
- BEFOR เมื่อกดเลือกรายการนี้ โปรแกรมจะถอยหลัง 1 PART
- EDIT เมื่อกดเลือกรายการนี้ โปรแกรมจะเข้าไปแก้ไขข้อมูลในรายการขณะนั้น
- ADD เมื่อกดเลือกรายการนี้ โปรแกรมจะเพิ่มรายการข้อมูลใหม่
- DELETE เมื่อกดเลือกรายการนี้ โปรแกรมจะลบข้อมูลในรายการนั้นออก
- QUIT เมื่อกดเลือกรายการนี้ โปรแกรมจะออกจากเมนูหัวข้อนี้

และตรงทางเลือก OPTION การเพิ่มเติมข้อมูล ถ้าเป็นเครื่องปรับอากาศ แบบ

- 1) ใช้ HEATER และชิ้นงานนี้ใช้เฉพาะ HEATER เดิม OPTION : H
- 2) ใช้ CONTROL BOX 'C' และชิ้นงานนี้ใช้เฉพาะ CONTROL BOX C เดิม OPTION : C
- 3) ใช้ CONTROL BOX + HEATER และชิ้นงานนี้ใช้เฉพาะ CONTROL+HEATER เดิม OPTION : CH
- 4) ใช้ BACK PANEL BV และชิ้นงานนี้ใช้เฉพาะ BACK PANEL BV เดิม OPTION : B
- 5) ใช้ BACK PANEL PV และชิ้นงานนี้ใช้เฉพาะ BACK PANEL PV เดิม OPTION : P
- 6) ใช้ BACK PANEL OS และชิ้นงานนี้ใช้เฉพาะ BACK PANEL OS เดิม OPTION : O
- 7) ใช้ BACK PANEL SS และชิ้นงานนี้ใช้เฉพาะ BACK PANEL SS เดิม OPTION : S
- 8) ใช้ ROTARY และชิ้นงานนี้ใช้เฉพาะ ROTARY เดิม OPTION : R

- 9) ปกติที่มี OPTION และ ไม่มี OPTION ใช้ชิ้นงานธรรมดา เติม OPTION : N
- 10) ปกติที่มี OPTION และ ไม่มี OPTION ใช้ชิ้นงานแบบมีตัวเลือก เติม OPTION : Z

จ. หมวด PLAN ใช้ในการวางแผนการจัดลำดับงาน และ ออกรายงานมีอยู่ 3 รายการ ให้เลือกโดยลูกศร ขึ้นลง ดังนี้ เมื่อเลือกเสร็จจะออก กด ESC ดังรูปที่ 6.14

SETUP	TRANSACTION	VIEW	INTERNAL	PLAN	EXIT
					PART USAGE JOB SCHEDULING -FIND TOTAL PART -STOCK BALANCE -CUT PROCESS -PUNCH PROCESS -PRESS BAKE -ARC & SOLDER

รูปที่ 6.14 แสดงหน้าจอในการใช้งานหมวด PLAN ของโปรแกรม

1. รายการ PART USAGE PER ORDER ใช้หาจำนวนชิ้นงานที่ต้องการใน 1คำสั่งผลิตแต่ละใบ
 ดังรูปที่ 6.15

JOB #	CODE	DATE REQUIRE	QTY (SETS)	CUSTOMER	TYPE MODEL
JB001	7402J01	02/06/92	100	UNI-AIRE	MFC 1600 COOL ONLY

ID	CODE	PART NAME	PCS/SET	PCS/PO
1	6402J01	FRAME FILTER (1600)	1	100
2	6402J04	VENTURY (1600)	1	100
3	6402J06	BACK PANEL (1600)	1	100
4	6402J19	DRAIN PAN	1	100

press any key to continue

FIND NEXT BEFOR PRINT	QUIT
-----------------------	------

รูปที่ 6.15 แสดงหน้าจอในการใช้งานรายการ PLAN PART USAGE PER ORDER ของโปรแกรม

2. JOB SCHEDULING ต้องป้อน ข้อมูลเกี่ยวกับจำนวน เครื่องจักร คนงาน ก่อนวางแผน โดยต้องเลือกงานที่ต้องการวางแผน ได้ไม่เกิน 8 งาน ต่อการวางแผนแต่ละครั้ง ก่อนการวางแผนต้องเลือก การเรียงลำดับ ซึ่งสามารถเลือกได้ 2 วิธีคือ การเรียงตามรหัสชิ้นงานและการเรียงตามลำดับความสำคัญในการผลิตก่อนหลัง จากนั้นก็ป้อนข้อมูลตามแต่ที่โปรแกรมต้องการ เช่น จำนวนคนงานที่มี , จำนวนเครื่องจักรที่มี , จำนวนชั่วโมงการทำงาน เป็นต้น ซึ่งโปรแกรมจะทำการคำนวณค่าที่ได้ และแผนงานการจัดลำดับงานการป้อนลง เครื่องจักรได้ รายงานดังนี้

- ให้รายงานการหาจำนวนชิ้นส่วนที่ต้องใช้รวมของทุกงานในช่วง วันเวลาของการวางแผน
- ให้รายงานการหายอดที่ตัดยอดแล้ว จากที่คงคลังก่อนจะนำยอดนี้มาวางแผนก่อนการผลิต
- ให้รายงานการหากำลังเครื่องจักร (MACHINE-HR)กำลังคน (MAN-HR) เปอร์เซนต์การใช้งานเครื่องจักร การตั้งเครื่องจักร และการจัดลำดับงาน ของหน่วยผลิต ตัด
- ให้รายงานการหากำลังเครื่องจักร (MACHINE-HR)กำลังคน (MAN-HR) เปอร์เซนต์การใช้งานเครื่องจักร การตั้งเครื่องจักร และการจัดลำดับงาน ของหน่วยผลิต บีม
- ให้รายงานการหากำลังเครื่องจักร (MACHINE-HR)กำลังคน (MAN-HR) เปอร์เซนต์การใช้งานเครื่องจักร การตั้งเครื่องจักร และการจัดลำดับงาน ของหน่วยผลิต พับ
- ให้รายงานการหากำลังเครื่องจักร (MACHINE-HR)กำลังคน (MAN-HR) เปอร์เซนต์การใช้งานเครื่องจักร และการจัดลำดับงาน ของหน่วยผลิต อาร์ค และ ตกแต่ง ดังรูปที่ 4.15 ถึง รูปที่ 4.19 ข้างต้น

จ. หมวด EXIT ใช้สำหรับการออกจากโปรแกรมเพื่อไป โปรแกรม Foxpro หรือไป ระบบ DOS

SETUP	TRANSACTION	VIEW	INTERNAL	PLAN	EXIT
					1
					EXIT TO DOT PROMPT EXIT TO DOS
DO YOU WANT TO EXIT (YES/NO) :					

รูปที่ 6.16 แสดงหน้าจอการใช้งานรายการ EXIT

1. EXIT TO DOT PROMPT ออกไปสู่ DOT PROMPT ยังอยู่ในโปรแกรม Foxpro
2. EXIT TO DOS ออกจากโปรแกรมที่ทำงาน และโปรแกรม Foxpro ไปสู่ DOS

3. การวิเคราะห์ข้อมูล ในที่นี้จะมียางานจากการวางแผนออกมา 5 แบบ

ในการวิเคราะห์ข้อมูลมี 6 ด้านที่ต้องพิจารณาในแง่กำลังการผลิต และการวางแผนดังนี้

- พิจารณาคอขวด M/C (MIN) และ MAN (MIN) ว่าใช้เวลาในการทำงานเหมาะสมหรือไม่ โดยดูว่าเวลาที่ได้จากรายงานการจัดลำดับงานนี้ ว่ามีค่ามากหรือน้อยไปอย่างไร สามารถนำไปแก้ไขให้ทันกาล (UPDATE) ได้ในส่วนของการ PROCESS TIME ของชิ้นงาน โดยต้องป้อนค่าเป็น นาที ต่อ ชิ้นงาน และหากหาเวลามาตรฐานในการผลิตชิ้นงาน และเวลาตั้งเครื่องจักรได้ทุกรหัสชิ้นงาน ก็จะทำให้ในการวางแผนครั้งต่อไปจะได้สามารถ ตั้ง เป็น มาตรฐานการทำงานได้ จนถึงอาจมีการให้รางวัล พนักงาน (Incentive Money) ที่ทำงานได้เกินมาตรฐาน
- พิจารณาคอขวด M/C จะพบว่ามีสัญลักษณ์ของเครื่องจักร กักขังอยู่นั้นคือ แถวชิ้นงานในแถว นั้น ควรจะผ่านเครื่องจักรที่แสดงในสัญลักษณ์ นั้นๆ ซึ่งเป็นการช่วยให้รู้ว่า ชิ้นงานนั้นๆ จะต้องผ่านเครื่องจักรใดบ้างจะช่วยในการ เตรียมแผนการล่วงหน้าตั้ง เครื่องจักรให้เหมาะกับงานที่จะต้องทำ
- พิจารณา แถวชิ้นงานที่เรียงลำดับลงไปจะเป็นการเรียงตามลำดับ ความสำคัญก่อนหลังในการนำชิ้นงานนั้นเข้าทำการผลิต ซึ่งจะช่วยให้การใส่งานที่เหลือ เป็นไปด้วยความสะดวก เพราะว่าชิ้นงานที่สำคัญจะต้องได้รับการผลิตก่อน ชิ้นงานที่มีความสำคัญน้อย และก็จะทำให้ชิ้นงานที่เสร็จช้าส่วนใหญ่เป็นชิ้นงานที่มีความสำคัญน้อย กับการประกอบเครื่องปรับอากาศ
- เปรียบเทียบ MAN - HR GIVE , M/C-HR GIVE กับ MAN - HR USE , M/C - HR USE ถ้าหาก จำนวนชั่วโมง คนงานที่ใช้ น้อยกว่า จำนวน ชั่วโมงคนงานที่ใช้ อาจจะทำให้การทำงานในการวางแผนนั้นทำไม่ทัน แต่หาก มีจำนวนชั่วโมงคนงานที่ใช้ มากกว่า จำนวนชั่วโมงคนงานที่ใช้ ก็ควรที่จะทำงานในการวางแผนนั้น ๆ ได้ ซึ่งในจุดนี้จะช่วยให้ผู้วางแผนงานนั้นแผนกขึ้นรูปชิ้นงานโลหะแผ่น สามารถมองเห็นภาพรวมในการที่จะจัดสรรพนักงาน และแบ่งงานให้แต่ละจุดผลิตย่อยอย่างมีความเป็นเหตุผลและสมคูลมากขึ้น
- พิจารณาคู WORK TIME และ SETUP TIME ของแต่ละเครื่องจักร หากพบว่าตัวคีมงานมากไป ก็สามารถลองพิจารณา แบ่งงานไปให้กับเครื่องจักรที่ยังค่อนข้างว่างอยู่ได้ เป็นการจัดสมดุลของงานให้มีประสิทธิภาพมากยิ่งขึ้นได้ (แต่ทั้งนี้ ต้องพิจารณาดูด้วยว่า เครื่องจักรที่จะนำงานไปแทนนั้น ทำชิ้นงานนั้นได้หรือไม่ และ กำลังคนงานมีเพียงพอที่จะทำงานในการวางแผนนั้นๆ ได้หรือไม่
- พิจารณา MAN CAPACITY SHOULD APPROX ว่าจำนวนคนงานที่เหมาะสมในการทำงานครั้ง

ที่วางแผนนี้ มีความเหมาะสมในแง่กำลังการผลิตหรือไม่ โดยในการคำนวณได้เพื่อ การ
ขนย้ายในการทำงานประมาณ 10 % หากค่าที่ได้มีมากกว่าคนงานในปัจจุบันและคนงานทำการ
ผลิตได้ทัน แสดงว่าข้อมูลเวลาภายในโปรแกรมเข้าใบควรแก้ไขให้เร็วขึ้น แต่หากค่าที่ได้เท่า
กับปัจจุบัน และคนงานทำการผลิตทันแสดงว่า ข้อมูลเวลาในโปรแกรมใกล้เคียงค่าเวลาผลิต
จริง ส่วนถ้าหากค่าที่ได้มีน้อยกว่าคนงานในปัจจุบัน แต่จำนวนคนงานปัจจุบันทำการผลิตขึ้นงาน
ทันแสดงว่า ข้อมูลเวลาในการผลิตในโปรแกรมเร็วไป ควรปรับแก้ไขให้ช้าลง

- แต่หากมีความสงสัยในการวางแผน การจัดทำตลับงานนี้ สามารถตรวจสอบแก้ไขปรับปรุง
(INTERNAL DATA) เกี่ยวกับเวลา ในการตั้ง เครื่องจักร และ เวลาในการทำงานได้จาก
ข้อมูลในโปรแกรมได้ตลอดเวลา)

4.การเพิ่มเติมผลิตภัณฑ์ และการเตรียมข้อมูลที่จำเป็นสำหรับโปรแกรม

ในการเพิ่มเติมข้อมูลผลิตภัณฑ์ใหม่เข้าโปรแกรม เพื่อขยายขีดความสามารถในการทำงานให้มากขึ้น
นั้นจำเป็นต้องหาข้อมูลมาสนับสนุนดังนี้

4.1 ข้อมูลเพื่อทำรายงานการทำงานชิ้นงานที่ใช้ต่อใบสั่งผลิต

ข้อมูลแบบ TRANSACTION

- ข้อมูลใบสั่งผลิต เช่น เบอร์งานที่ใช้ (Job No.) รหัสผลิตภัณฑ์ (Product Code)
วัน-เดือน-ปี ที่ต้องการวางแผน และ จำนวนชุดที่ต้องการ

ข้อมูลแบบ INTERNAL

- ข้อมูลเกี่ยวกับผลิตภัณฑ์แอร์ เช่น ต้องบ่อน รหัสผลิตภัณฑ์แอร์ใหม่ลงใบพร้อมทั้งรายละเอียดข้อมูล
เกี่ยวกับแอร์ที่ต้องการทราบ เช่น ใช้อุปกรณ์ไฟฟ้าอะไร , ใช้คอยล์แบบไหน , สีอะไร และ
ใครคือลูกค้า เป็นต้น
- ข้อมูลเกี่ยวกับรายละเอียดของชิ้นงาน เช่น ใช้รหัสชิ้นงานอะไร , ขนาดเท่าไร , เบอร์รูปภาพ
อะไร และใช้วัสดุอะไรในการทำงานนั้นๆ เป็นต้น
- ข้อมูลเกี่ยวกับสูตรการผลิต เช่น ชิ้นงานรหัสนี้ใช้กับแอร์รุ่นอะไร , มีค่าความสำคัญเท่าไร , ค่า
ตัวเลือกเป็นอย่างไร และใช้กี่ชิ้นงานต่อแอร์ 1 ชุด เป็นต้น

4.2 ข้อมูลเพื่อทำรายงานการตัดยอดชิ้นงานในพัสดุคงคลัง

ข้อมูลแบบ TRANSACTION

- ข้อมูล วัน-เดือน-ปี , รหัสชิ้นงาน และจำนวน ที่ชิ้นงานเข้า พักคงคลัง
- ข้อมูล วัน-เดือน-ปี , รหัสชิ้นงาน และจำนวน ที่ชิ้นงานออก พักคงคลัง

4.3 ข้อมูลใช้ทำรายงานการจัดลำดับงานและประมาณกำลังการผลิตในหน่วยผลิตย่อย

ข้อมูลแบบ TRANSACTION

- ข้อมูลเวลาเป็นชั่วโมงการทำงานที่มีในแต่ละจุดการผลิต พร้อมทั้งจำนวนเครื่องจักรและคนงานที่สามารถทำงานได้ สำหรับการวางแผนในแต่ละครั้งของการวางแผน

ข้อมูลแบบ INTERNAL

- ข้อมูลเวลาเครื่องจักรในการผลิตชิ้นงานต่างๆ และการตั้งเครื่องจักรของแต่ละรหัสชิ้นงานในหน่วยผลิตย่อย ตัด -> บีม -> พับ -> อาร์ค-ตกแต่ง พร้อมจำนวนคนงานที่ทำงานกับเครื่องจักรในสภาวะปกติ

5. ตัวอย่างการเพิ่มเติมผลิตภัณฑ์

ในการเพิ่มเติมผลิตภัณฑ์ มีอยู่ 2 แบบที่ต้องคำนึงถึงคือ แบบที่ 1 มีข้อมูลผลิตภัณฑ์เดิมอยู่แล้วพร้อมข้อมูลรหัสชิ้นงาน , เวลาในการผลิต - ตั้งเครื่องจักร และต้องการเพิ่มรหัสผลิตภัณฑ์ใหม่เข้าไป และแบบที่ 2 ไม่มีข้อมูลผลิตภัณฑ์และรหัสชิ้นงานเดิมอยู่เลย ต้องสร้างรหัสผลิตภัณฑ์และรหัสชิ้นงาน พร้อมทั้ง เวลาในการผลิต - ตั้งเครื่องจักร ใหม่หมด

แบบที่ 1 เพิ่มเติมข้อมูลรหัสผลิตภัณฑ์ใหม่โดยมีรหัสผลิตภัณฑ์เดิมอยู่แล้ว

เช่น เพิ่มเติมรหัสผลิตภัณฑ์ 7402N75 เครื่องปรับอากาศในห้อง มีรายละเอียดผลิตภัณฑ์ดังนี้
 UE = MFC , APPLICATION = COOL ONLY , VAL = B/B , THERMO = "C" + DOOR2 ,
 COLOR = U-12 , ELEC. SYSTEM = 24V /50/60 HZ , PACKING = EXPORT + FOAM ,
 TRANSFER = YES , SOCKET = NO , REMARK = 04721-015

และ เพิ่มเติมรหัสผลิตภัณฑ์ 7205E33 เครื่องปรับอากาศนอกห้อง มีรายละเอียดผลิตภัณฑ์ดังนี้
 UE = ACM , APPLICATION = COOL ONLY , FIN = AL , VAL = S/S , COMPRESSOR
 = TECUMSHE , HEATE = NO , COLOR = U-23 , ELEC. SYSTEM = 220/1/50 , PACKING
 = EXPORT , START.KIT = YES , HI & LO = NO , TIME = NO , REMARK = 04721-010

ขั้นที่ 1 หลังจากเข้าโปรแกรมการจัดลำดับงานแล้วให้ไปที่ รายการ INTERNAL จากนั้น เลือก รายการ PRODUCT TYPE OF AIR จากนั้นก็เลือกรายการด้านล่างของจอภาพโดยกด <A>dd ก็จะทำให้ปรากฏหน้าจอตั้งรูปที่ 6.17 และรูปที่ 6.18

AIR PRODUCT SPECIFICATION			
PRODUCT CODE :	CUSTOMER :	UE :	CM :
APPLICATION :		THERMO	
VAL :		TRANSFER	
ELEC.SYS :			
COLOR :		SOCKET	
PACKING :			OPTION :
REMARK :			

FIND	NEXT	BEFOR	TOP	LAST	EDIT	ADD	DELETE	PRINT	QUIT
------	------	-------	-----	------	------	-----	--------	-------	------

รูปที่ 6.17 แสดงหน้าจอรายละเอียดผลิตภัณฑ์ที่พร้อมจะป้อนข้อมูลแบบ MFC

AIR PRODUCT SPECIFICATION			
PRODUCT CODE :	CUSTOMER :	UE :	CM :
VAL :		FIN COIL :	HI & LO :
ELECTRIC SYS :		COMPRESSOR :	
COLOR :		HEATE :	TIMER :
PACKING :			OPTION :

FIND	NEXT	BEFOR	TOP	LAST	EDIT	ADD	DELETE	PRINT	QUIT
------	------	-------	-----	------	------	-----	--------	-------	------

รูปที่ 6.18 แสดงหน้าจอรายละเอียดผลิตภัณฑ์ที่พร้อมจะป้อนข้อมูลแบบ ACM

ขั้นที่ 2 ให้ป้อนข้อมูลลงไปตามรายการต่างๆ ที่ปรากฏในหน้าจอ โดยเมื่อป้อนในรายการใดเสร็จ ให้กด ENTER จนเสร็จทุกรายการ ข้อมูลก็จะเพิ่มเข้าไปในฐานข้อมูลเอง โดยในส่วนของ OPTION นั้น มีสัญลักษณ์ให้เติมดังนี้ H = HEATER , Z = NORMAL PART และถ้าเป็นแบบ MFC ก็จะมีสัญลักษณ์ C = CONTROL ส่วนถ้าเป็นแบบ ACM ก็เพิ่มสัญลักษณ์ B = BACK PANEL BV , P = BACK PANEL PV , O = BACK PANEL OS , S = BACK PANEL SS

เช่น ถ้าหากเป็นเครื่องปรับอากาศแบบ MFC รุ่น 7402N75 ก็เติม OPTION = CZ โดยสัญลักษณ์แทนเครื่องปรับอากาศแบบ มี CONTROL ไม่มี HEATER หรือว่าหากรู้ว่าเป็นเครื่องปรับอากาศแบบ

ไม่มี CONTROL แต่มี HEATER ก็ให้เติม OPTION = HZ และถ้าไม่มีทั้ง CONTROL และ HEATER
 ก็ให้เติม OPTION = Z ตัวเดียว ส่วนในเครื่องปรับอากาศแบบ ACM รุ่น 7205E33 เติม OPTION
 = SZ ซึ่งหมายถึงใช้ BACK PANEL SS และ ไม่มี HEATER แต่ถ้ารู้ว่า เป็นเครื่องปรับอากาศแบบ
 มี HEATER ใช้ BACK PANEL PV ก็ให้เติม OPTION = PH เป็นต้น ก็จะได้รูปที่ 6.19 และ รูป
 ที่ 6.20 แสดงตัวอย่างในการป้อนข้อมูลลงไป

AIR PRODUCT SPECIFICATION			
PRODUCT CODE	: 7402N75	CUSTOMER : CLASSIC	UE : MFC CM : MFC
APPLICATION	: COOL ONLY	THERMO	: "C" + DOOR2
VAL	: B/B	TRANFER	: YES
ELEC.SYS	: 24V /50/60 HZ	SOCKET	: NO
COLOR	: U-12	OPTION	: CZ
PACKING	: EXPORT + FOAM		
REMARK	: 04721 -015		
FIND NEXT BEFOR TOP LAST EDIT ADD DELETE PRINT QUIT			

รูปที่ 6.19 แสดงหน้าจอการป้อนรหัสผลิตภัณฑ์ 7402N75 เครื่องปรับอากาศในท้อง MFC

AIR PRODUCT SPECIFICATION			
PRODUCT CODE	: 7205E33	CUSTOMER : UNI - AIRE	UE : ACM CM : ACM
VAL	: S/S	FIN COIL	: AL HI & LO : NO
ELECTRIC SYS	: 220/1/50	COMPRESSOR	: TECUMSHE
COLOR	: U-23	HEATE	: NO TIMER : NO
PACKING	: EXPORT	OPTION	: SZ
FIND NEXT BEFOR TOP LAST EDIT ADD DELETE PRINT QUIT			

รูปที่ 6.20 แสดงหน้าจอการป้อนรหัสผลิตภัณฑ์ 7205E33 เครื่องปรับอากาศนอกห้อง ACM

แบบที่ 2 เพิ่มเติมข้อมูลรหัสผลิตภัณฑ์ใหม่โดยไม่มีรหัสผลิตภัณฑ์เดิมอยู่แล้ว

ขั้นที่ 1 ต้องไปป้อนข้อมูลรายละเอียดชิ้นงานโดยเลือกรายการ INTERNAL และมาเลือกรายการ PART SPECIFICATION ในหน้าจอตั้งรูปที่ 6.21 โดยต้องป้อนข้อมูลชิ้นงานทุกชิ้นงานของผลิตภัณฑ์นั้นๆ เพราะว่าโปรแกรมการจัดลำดับงาน จะหาชิ้นงานเท่าที่หาพบในฐานข้อมูลเท่านั้น

PART SPECIFICATION				
CODE	: 6205S02	NAME	: BOTTOM PLATE (12-36)	
MATERIAL	: EG # 19	MODEL	USE/SET	PRIORITY
		10		2
SIZE (mm *mm)	: 986*396	11		
		12	1	
DRAWING	: 4647002	15	1	
		16	1	
TURRET PROG.	: 231	18	1	
		20	1	
LVD PRG	:	25	1	
		30	1	
AMADA PRG	:	36	1	
		50		
		60		

FIND NEXT BEFOR TOP LAST EDIT ADD DELETE PRINT QUIT

รูปที่ 6.21 หน้าจอแสดงรายละเอียดชิ้นงานพร้อมที่จะป้อนข้อมูลชิ้นงานลงไป

ขั้นที่ 2 ใส่เวลาในการผลิตชิ้นงาน โดยเลือกรายการ INTERNAL และมาเลือกรายการ PROCESS TIME ดังในรูปที่ 6.22 โดยค่าเวลาที่ใส่มีหน่วยเป็น นาที หมดทุกรายการ

FABRICATE PROCESS

CODE :

6205P02

BOTTOM PLATE (10-11)

SHEAR TIME MAN PRESS BAKE MAN

LVD : 0.30 2 LVD :0.00 2
 S023 : 0.30 AMADA:0.40
 S024 : 0.00 KLEEN: 0.00
 CHAI :0.00
 PUNCH TIME 24 " :0.45

TURRET :0.60 ARC & SOLDER TIME
 REXROT :0.00 SPOT W :2.00
 UNIPUN :0.00 SUR FI. :2.50
 SMALL :0.00 ARC :0.40
 (1,2,3,4) SOLDER :5.50
 MEDIUM :0.24
 (5,6)
 LARGE :0.00
 (7,8,9,10,11,12,13)

SETUP TIME (MIN)

SHEAR GROUP

LVD : 5.00
 S023 : 7.00
 S024 : 0.00

PUNCH GROUP

TURRET :15.00 SMA :0.00
 REXROT :0.00 MED :10.00
 UNIPUN :0.00 LAR :0.00

PRESS BAKE GROUP

LVD :0.00
 AMADA :15.00
 KLEEN :0.00
 CHAI :0.00
 24 " :15.00

FIND NEXT BEFOR TOP

LAST EDIT ADD DELETE PRINT QUIT

รูปที่ 6.22 หน้าจอแสดงรายละเอียดเวลาการผลิตชิ้นงานพร้อมที่จะป้อนข้อมูลชิ้นงานลงไป

ขั้นที่ 3 ใส่ข้อมูลสูตรการผลิตจากการเลือกรายการ INTERNAL และมาเลือกรายการ FORMULAR OF PRODUCTION หน้าจอแสดงดังในรูปที่ 6.23

CODE	: 6205P02	NAME	: BOTTOM PLATE (10-12)
MODEL	USE./SET	OPTION	OPTION MEAN
10;A	1	Z	N = NORMAL PART
11;B	1	Z	C = CONTROL BOX 'C' PART
12;C			H = HEATER PART
15;D			CH = CONTROL + HEATER PART
16;E			B = BACK PANEL BV
18;F			P = BACK PANEL PV
20;G			O = BACK PANEL OS
25;H			S = BACK PANEL SS
30;L			Z = CHOICE PART
36;J			R = ROTARY PART
50;N			
60;O			
FIND NEXT BEFOR TOP LAST EDIT ADD DELETE PRINT QUIT			

รูปที่ 6.23 แสดงหน้าจอในการที่จะป้อนรายละเอียดสูตรการผลิต

ขั้นที่ 4 ป้อนข้อมูลผลิตภัณฑ์เหมือนในแบบที่ 1 ข้างต้น 2 ขั้นตอน

จากนั้นโปรแกรมการจัดลำดับงาน ก็พร้อมที่จะทำงานได้ โดยทั้งนี้ความถูกต้องของโปรแกรมจะมีมากน้อยเพียงใด ขึ้นอยู่กับการใส่รายละเอียดข้อมูลต่างๆ ลงไปในฐานข้อมูล

ภาคผนวก จ

ชุดคำสั่งภาษาต้นฉบับ

*PROGRAM FOR JOB SCHEDULING OF FABRICATE LINE

*MAIN MENU PROGRAM

*BY MR.KIJJA TANGKITIWONGPORN C 315-892

*START 16 OCT 1991

*LAST EDITED 24/6/92

*M.ENG (IE) CHULALONGKORN UNIVERSITY

CLEAR ALL

RELEASE WINDOW ALL

SET TALK OFF

SET MESSAGE TO 23 CENTER

SET DEVICE TO SCREEN

CLEAR

@ 12,16 SAY 'PLEASE WAIT A MINUTE WHILE SETUP PARAMETER '

-----< DEFINE PUBLIC VARIABLE >-----

pub_print = SET("PRINT")

pub_drive = " "

pub_exit = .F.

pub_safe = SET('SAFETY')

pub_status = SET('STATUS')

pub_yn = .F.

-----< DEFINE MAIN MENU BAR >-----

DEFINE MENU main

DEFINE PAD setup OF main PROMPT 'SETUP' AT 1,1

DEFINE PAD trans OF main PROMPT 'TRANSACTION' AT 1,13

DEFINE PAD view OF main PROMPT 'VIEW' AT 1,30

DEFINE PAD inter OF main PROMPT 'INTERNAL' AT 1,38

DEFINE PAD plan OF main PROMPT 'PLAN' AT 1,50

```

DEFINE PAD report      OF main PROMPT 'REPORT'      AT 1,58
DEFINE PAD exit        OF main PROMPT 'EXIT'        AT 1,68
ON SELECTION PAD setup OF main ACTIVATE POPUP pop_setup
ON SELECTION PAD trans OF main ACTIVATE POPUP pop_trans
ON SELECTION PAD view  OF main ACTIVATE POPUP pop_view
ON SELECTION PAD inter OF main ACTIVATE POPUP pop_inter
ON SELECTION PAD plan  OF main ACTIVATE POPUP pop_plan
ON SELECTION PAD report OF main ACTIVATE POPUP pop_report
ON SELECTION PAD exit  OF main ACTIVATE POPUP pop_exit

*-----< DEFINE FILE POPUP SETUP MENU >-----*

DEFINE POPUP pop_setup FROM 3,0

DEFINE BAR 1 OF pop_setup PROMPT 'SPECIFY PATH FOR DATABASE'
DEFINE BAR 2 OF pop_setup PROMPT '_____ ' SKIP
DEFINE BAR 3 OF pop_setup PROMPT 'BACKUP RECORD FILE TO DISK ';
+'CONFIRM BACKUP RECORD FILE TO FLOPPY DISK OR HARD DISK'
DEFINE BAR 4 OF pop_setup PROMPT '_____ ' SKIP
DEFINE BAR 5 OF pop_setup PROMPT 'ERASE FILE FROM DISK ';
+'CONFIRM FINISH CURRENT PURCHASE ORDER TO RECORD'
DEFINE BAR 6 OF pop_setup PROMPT '_____ ' SKIP
DEFINE BAR 7 OF pop_setup PROMPT 'VIEW A PROGRAM ';
+'TO SEE PROGRAM BUT SHOULD NOT EDIT PROGRAM '
DEFINE BAR 8 OF pop_setup PROMPT '_____ ' SKIP
DEFINE BAR 9 OF pop_setup PROMPT 'TO SYSTEM INFORMATION ';
+'SHOW CURRENT SYSTEM INFORMATION ON YOUR COMPUTER '
DEFINE BAR 10 OF pop_setup PROMPT '_____ ' SKIP
DEFINE BAR 11 OF pop_setup PROMPT 'REINDEX ALL FILE ';
+'REINDEX INTERNAL FILE TO WHEN UPDATE INTERNAL DATA'

ON SELECTION POPUP pop_setup DO setup_proc

*-----< DEFINE POPUP TRANSACTION MENU >-----*

```

```

DEFINE POPUP pop_trans FROM 3,13
DEFINE BAR 1 OF pop_trans PROMPT 'MANUFACTURING ORDER 'MESS SPACE(5);
+'INPUT DETAIL OF PURCHASE ORDER MFC & ACM MODEL '
DEFINE BAR 2 OF pop_trans PROMPT 'STOCK BALANCE' MESS SPACE(5);
+'BALANCING STOCK FROM INPUT & OUTPUT PART'
DEFINE BAR 3 OF pop_trans PROMPT '_____ 'SKIP
DEFINE BAR 4 OF pop_trans PROMPT 'STOCK INPUT' MESS SPACE(10);
+'INPUT ACTUAL PRODUCTION DATA TO STORE'
DEFINE BAR 5 OF pop_trans PROMPT 'STOCK OUTPUT' MESS SPACE(10);
+'OUTPUT ACTUAL PRODUCTION DATA FROM STORE'
ON SELECTION POPUP pop_trans DO trans_proc
*-----< DEFINE POPUP VIEW MENU >-----*
DEFINE POPUP pop_view FROM 3,30
DEFINE BAR 1 OF pop_view PROMPT 'M A S T E R   D A T A 'SKIP
DEFINE BAR 2 OF pop_view PROMPT '_____ 'SKIP
DEFINE BAR 3 OF pop_view PROMPT ' MANUFACTURING ORDER ' MESS SPACE(10);
+'TO SEE CURRENT PURCHASE ORDER'
DEFINE BAR 4 OF pop_view PROMPT '   STOCK   INPUT       ' MESS SPACE(10);
+'TO SEE OVERALL PART INPUT '
DEFINE BAR 5 OF pop_view PROMPT '   STOCK   OUTPUT       ' MESS SPACE(10);
+'TO SEE OVERALL PART OUTPUT'
DEFINE BAR 6 OF pop_view PROMPT '   STOCK   BALANCE     ' MESS SPACE(10);
+'TO SEE OVERALL BALANCE STOCK '
DEFINE BAR 7 OF pop_view PROMPT '_____ 'SKIP
DEFINE BAR 8 OF pop_view PROMPT 'I N T E R N A L   D A T A 'SKIP
DEFINE BAR 9 OF pop_view PROMPT '_____ 'SKIP
DEFINE BAR 10 OF pop_view PROMPT ' PRODUCT TYPE OF AIR 'MESS SPACE(10);
+'TO SEE CURRENT PRODUCT TYPE ACM & MFC OF AIR'
DEFINE BAR 11 OF pop_view PROMPT ' SHEET METAL PART OF AIR 'MESS SPACE(10);

```

```

+'TO SEE CURRENT SHEET METAL PART USED OF AIR IN PRODUCTION'
DEFINE BAR 12 OF pop_view PROMPT ' PROCESS OF SHEET METAL 'MESS SPACE(10);
+'TO SEE PROCESS TIME IN EACH STATION OF EACH PART'
DEFINE BAR 13 OF pop_view PROMPT ' FORMULATION OF PRODUCTION' MESS SPACE(10);
+'TO SEE PART CODE IS USED PER MODEL PER TYPE OF AIR'
DEFINE BAR 14 OF pop_view PROMPT ' PURCHASE PART ' MESS SPACE(10);
+'TO SEE PURCHASE PART SPECIFICATION'
DEFINE BAR 15 OF pop_view PROMPT ' FORMULAR PURCHASE PART ' MESS SPACE(10);
+'TO SEE FORMULATION PURCHASED PART'
ON SELECTION POPUP pop_view DO view_proc
*-----< DEFINE POPUP INTERNAL MENU >-----*
DEFINE POPUP pop_inter FROM 3,38
DEFINE BAR 1 OF pop_inter PROMPT 'SHEET METAL OF PART 'MESS SPACE(10);
+'GIVE DATA SPECIFICATION OF PART MFC AND ACM'
DEFINE BAR 2 OF pop_inter PROMPT '_____ 'SKIP
DEFINE BAR 3 OF pop_inter PROMPT 'TYPE OF PRODUCT AIR 'MESS SPACE(10);
+'GIVE DATA SPECIFICATION OF AIR PRODUCT '
DEFINE BAR 4 OF pop_inter PROMPT '_____ 'SKIP
DEFINE BAR 5 OF pop_inter PROMPT 'PROCESS TIME EACH PART'MESS SPACE(10);
+'GIVE DATA PROCESS TIME OF PART ACM & MFC'
DEFINE BAR 6 OF pop_inter PROMPT '_____ 'SKIP
DEFINE BAR 7 OF pop_inter PROMPT 'FORMULATION OF PRODUCTION'MESS SPACE(10);
+'GIVE DATA FORMULATION OF ASSEMBLY IN PRODUCTION'
ON SELECTION POPUP pop_inter DO inter_proc
*-----< DEFINE POPUP PLAN MENU >-----*
DEFINE POPUP pop_plan FROM 3,50
DEFINE BAR 1 OF pop_plan PROMPT ' PART USAGE PER PO'MESS SPACE(10);
+'FIND ALL PART WHICH BE USED BY JOB ORDER NO.'
DEFINE BAR 2 OF pop_plan PROMPT '_____ 'SKIP

```



```

DEFINE BAR 3 OF pop_plan PROMPT 'JOB SCHEDULING ' MESS SPACE(10);
+'JOB SCHEDULING OF EACH PART TO FABRICATE + MAN -HR ,MACHINE - HR USED'
DEFINE BAR 4 OF pop_plan PROMPT '- FIND TOTAL PART 'SKIP
DEFINE BAR 5 OF pop_plan PROMPT '- STOCK BEFORE PLAN 'SKIP
DEFINE BAR 6 OF pop_plan PROMPT '- CUT PROCESS 'SKIP
DEFINE BAR 7 OF pop_plan PROMPT '- PUNCH PROCESS 'SKIP
DEFINE BAR 8 OF pop_plan PROMPT '- PRESS BAKE PROCESS 'SKIP
DEFINE BAR 9 OF pop_plan PROMPT '- ARC & SOLDER PROCESS'SKIP
DEFINE BAR 10 OF pop_plan PROMPT '-----'SKIP
DEFINE BAR 11 OF pop_plan PROMPT 'AUTO PLAN & REPORT 'MESS SPACE(10);
+ 'JOB PLAN ON LINE BY USE DATE START AND DATE FINISH PLAN'
DEFINE BAR 12 OF pop_plan PROMPT 'PURCHASED PART 'MESS SPACE(10);
+ 'JOB PLANNING ON PURCHASED PART OF AIR BEFORE ASSEMBLY'
ON SELECTION POPUP pop_plan DO plan_proc
*-----< DEFINE POPUP REPORT MENU >-----*
DEFINE POPUP pop_report FROM 3,58
DEFINE BAR 1 OF pop_report PROMPT 'INTERNAL DATA 'SKIP
DEFINE BAR 2 OF pop_report PROMPT '- PURCHASE ORDER 'MESS SPACE(10);
+ 'LIST DATA OF CURRENT JOB ORDER'
DEFINE BAR 3 OF pop_report PROMPT '- PART OF AIR 'MESS SPACE(10);
+ 'LIST DATA OF PART AIR '
DEFINE BAR 4 OF pop_report PROMPT '- PRODUCT OF AIR 'MESS SPACE(10);
+ 'LIST DATA OF PRODUCT AIR'
DEFINE BAR 5 OF pop_report PROMPT '-----' SKIP
DEFINE BAR 6 OF pop_report PROMPT 'INVENTORY DATA ' SKIP
DEFINE BAR 7 OF pop_report PROMPT '- STOCK INPUT 'MESS SPACE(10);
+ 'LIST DATA OF PART INPUT TO STORE'
DEFINE BAR 8 OF pop_report PROMPT '- STOCK OUTPUT 'MESS SPACE(10);
+ 'LIST DATA OF PART OUTPUT FROM STORE'

```

```

DEFINE BAR 9 OF pop_report PROMPT '- STOCK BALANCE 'MESS SPACE(10);
+ 'LIST DATA OF PART BALANCE FROM STORE'
ON SELECTION POPUP pop_report DO report_proc
*-----< DEFINE POPUP EXIT MENU >-----*
DEFINE POPUP pop_exit FROM 3,68
DEFINE BAR 1 OF pop_exit PROMPT 'TO DOT PROMPT'MESS SPACE(10);
+'EXIT PROGRAM AND GO INTO DOT PROMPT'
DEFINE BAR 2 OF pop_exit PROMPT 'TO DOS 'MESS SPACE(10);
+'EXIT PROGRAM AND GO INTO DOS COMMAND'
ON SELECTION POPUP pop_exit DO exit_proc
*-----< DEFINE POPUP SUBLEVEL EDIT RECORD >-----*
DEFINE POPUP pop_edit FROM 9,29
DEFINE BAR 1 OF pop_edit PROMPT 'MODIFY RECORD'
DEFINE BAR 2 OF pop_edit PROMPT '_____','SKIP
DEFINE BAR 3 OF pop_edit PROMPT 'UPDATE MASTER'
ON SELECTION POPUP pop_edit DO edit_proc
*-----< DEFINE OTHER WINDOW>-----*
DEFINE WINDOW working FROM 1,2 TO 18,35 FLOAT COLOR SCHEME 1
DEFINE WINDOW clr_scr FROM 0,0 TO 24,79 NONE
*-----< DEFINE OTHER POPUP>-----*
DEFINE POPUP getpart FROM 3,5 TO 20,45 COLOR SCHEME 4 PROMPT FIELD name ;
MESS 'PRESS AROOW TO SELECT AND [ENTER] TO DECISION'
ON SELECTION POPUP getpart DEACTIVATE POPUP
DEFINE POPUP getpo FROM 2,5 TO 25,20 COLOR SCHEME 4 PROMPT FIELD job ;
MESS 'PRESS ARROW SELECT PO AND PRESS [ENTER] TO DECISION'
ON SELECTION POPUP getpo DEACTIVATE POPUP
*****
* MAIN ROUTINE PROGRAM
*****

```

```
drive = SYS(5)      &&    CURRENT DRIVE      C:,D: etc
direc = CURDIR() &&    CURRENT DIRECTORY FOXV2\THESIS etc

DO display

DO moniter

ACTIVATE WINDOW clr_scr

@ 0,0 TO 2,79

DO set_envi

DO fkey_off

DO dbase      && OPEN DATABASE THAT CONCERN TO USE

DO WHILE .NOT. pub_exit

ACTIVATE MENU main

ENDDO

RELEASE WINDOW clr_scr

CLEAR

DO reset_envi

RETURN

*****

* PROCEDURE FILES

*****

*----< DEFINE AREA OF DATABASE >----*

PROCEDURE dbase

SELECT 1

USE part      INDEX part_cod  ALIAS partx

SELECT 2

USE product  INDEX prod_cod  ALIAS productx

SELECT 3

USE  used    INDEX used_cod  ALIAS usex

SELECT 4

USE po       INDEX po_date   ALIAS pox
```

```
SELECT 5
USE tra_po INDEX tpo_date ALIAS tpox
SELECT 6
USE stk_rec INDEX rec_date ALIAS stk_rx
SELECT 7
USE stk_shp INDEX shp_date ALIAS stk_sx
SELECT 8
USE stk_bal INDEX bal_date ALIAS stk_bx
SELECT 9
USE partpINDEX partp_cod ALIAS purx
SELECT 10
USE usep INDEX usep_cod ALIAS usez
RETURN
```

```
*---< MONITOR >---*
```

```
PROCEDURE moniter
```

```
IF ISCOLOR()
```

```
SET COLOR TO W/B,B/W
```

```
SET COLOR OF BOX TO W/B
```

```
ELSE
```

```
SET COLOR TO W/N,N/W
```

```
ENDIF
```

```
RETURN
```

```
*--< MONITOR INVERSE >----*
```

```
PROCEDURE inv_moni
```

```
IF ISCOLOR()
```

```
SET COLOR TO B/W,W+/B
```

```
ELSE
```

```
SET COLOR TO N/W,W/N
```

```
ENDIF
```



```
RETURN
*---< MONITOR BLINKING >---*
PROCEDURE blk_moni
IF ISCOLOR ()
SET COLOR TO W*/B,B/W
ELSE
SET COLOR TO W*/N,N/W
ENDIF
RETURN
*---< DISPLAY >---*
PROCEDURE display
@ 0,0 TO 24,79 CLEAR
DEFINE WINDOW show FROM 4,4 TO 20,65
ACTIVATE WINDOW SHOW
TEXT
```

U N I F A B E Q U I P M E N T C O . L T D ,

P R O D U C T I O N S C H E D U L I N G

F O R M F C & A C M M O D E L

Wriiten by Mr.Kijja Tangkitiwongporn ,M.IE,Chula ,1992

ENDTEXT

?SYS(2002)

KB = INKEY(10)

RELEASE WINDOW SHOW

?SYS(2002,1)

SET STATUS &pub_status

RETURN

--< SET_ENVI >--

PROCEDURE set_envi

SET ALTERNATE OFF

SET BELL OFF

SET BORDER TO SINGLE

SET CONFIRM ON

SET CONSOLEON

SET DATE TO BRITISH

SET DELETED OFF

SET DECIMALS TO 2

SET ESCAPE ON

SET EXACT ON

SET FIXED ON && MAKE 2 DECIMALS

SET SAFETY OFF && ON

SET STATUS OFF && ON

SET TALK OFF

SET ECHO OFF

SET HOURS TO 24

SET INTENSITY ON

SET AUTOSAVE ON && Flush Buffer To Disk

SET CENTURY OFF && Date 8 Digit

RETURN

---< RESET_ENVI >---

PROCEDURE reset_envi

SET BELL ON

SET CONFIRM OFF

SET CONSOLE OFF

SET DATE TO AMERICAN

```

SET DELETED OFF

SET EXACT OFF

SET SAFETY &pub_safe

SET STATUS &pub_status

SET PRINTER OFF

SET TALK ON

SET AUTOSAVE OFF

RETURN

*-----< DISABLE F1 - F10 KEYS >-----*

PROCEDURE fkey_off          && Set Function Key Off

I = 1

DO WHILE I <= 10

fn = 'F'+LTRIM(STR(I))

ON KEY LABEL &fn DISKEY = INKEY()

I = I+1

ENDDO

oldmode = INSMODE(.T.)    && Set Insert Mode On

oldlock = CAPSLOCK(.T.)  && Set Caplock On

oldlock = NUMLOCK(.F.)   && Set Numlock Off

RETURN

*----< msg2user >----*

PROCEDURE msg2user

PARAMETERS message1,message2,message3

IF message1 = 'CLOSE'

DEACTIVATE WINDOW msg2user

ELSE

ACTIVATE WINDOW msg2user

CLEAR

@ 0,0 SAY SPACE(1) + LEFT(message1 + SPACE(71),71) + SPACE(1)

```

```

@ 1,0 SAY SPACE(1);
+LEFT(IIF(PARAMETERS() < 2 ,SPACE(71),message2) +SPACE(71),71) +SPACE(1)
@ 2,0 SAY SPACE(1);
+ LEFT(IIF(PARAMETERS() < 3,SPACE(71),message3) + SPACE(71),71) + SPACE(1)
ENDIF
RETURN
*---< SETUP_PROC >---*
PROCEDURE setup_proc
DO CASE
CASE BAR() = 1
DEFINE POPUP dbfiles FROM 6,17 TO 18,36 PROMPT FILES LIKE *.DBF
ACTIVATE POPUP dbfiles
ON SELECTION POPUP dbfiles DO.opendb WITH PROMPT(),pub_drive
CASE BAR() = 3
dd      = SPACE(2)
filename = SPACE(14)
DEFINE WINDOW back FROM 13,12 TO 21,66
STORE GETFILE("DBF","File to Backup :") TO bkfi
IF .NOT. EMPTY(bkfi)
dbf_top = HEADER()
file_size = dbf_top + (RECSIZE() * RECCOUNT() +1 )
ACTIVATE WINDOW back
@ 1,0 SAY 'SOURCE FILE IS :' + '&bkfi'
@ 2,0 SAY 'WHAT DRIVE DO YOU WANT TO SELECT [A,B,C] : ' GET dd PICT '!: '
@ 3,0 SAY 'ENTER NAME DATABASE FILE TO BACKUP :' GET filename
READ
IF dd = 'A:' .OR. dd = 'B:'
DO backup WITH dd,bkfi,filename,file_size,drive,direc
ELSE

```



```
IF DISKSPACE() > 1.1 * (file_size * 3 )
@ 4,0 SAY 'ENOUGH DISKSPACE TO BACKUP FILE ON HARD DISK'
CLOSE DATA
USE &bkfi
COPY TO &dd&filename
@ 5,0 SAY 'BACKUP FILE COMPLETELY PRESS ANY KEY TO CONTINUE'
WAIT
CLOSE DATA
ELSE
@ 4,0 SAY 'NOT ENOUGH !! DISKSPACE TO BACKUP FILE'
ENDIF
ENDIF
DEACTIVATE WINDOW back
ELSE
WAIT 'THERE IS NO FILE TO BACKUP FILE'
CLOSE DATA
DEACTIVATE WINDOW back
ENDIF
DO dbase
CASE BAR() = 5
DEFINE WINDOW back FROM 13,12 TO 21,66
STORE GETFILE("DBF","File to Erase:") TO erfi
IF .NOT. EMPTY(erfi)
ACTIVATE WINDOW back
CLOSE DATA
ERASE &erfi
WAIT 'ERASE FILE COMPLETELY PRESS ANY KEY TO CONTINUE'
DEACTIVATE WINDOW back
ELSE
```

```
WAIT 'THERE IS NO FILE TO ERASE'  
CLOSE DATA  
DEACTIVATE WINDOW back  
ENDIF  
DO dbase  
CASE BAR() = 7  
STORE GETFILE("PRG","File to Edit :") TO edfi  
IF .NOT. EMPTY(edfi)  
= INSMODE(.T.)  
MODI COMM (edfi)  
ENDIF  
= INSMODE(.F.)  
CASE BAR() = 9  
DO sysinfo  
CASE BAR() = 11  
ACTIVATE WINDOW working  
CLOSE DATA  
SET TALK ON  
? 'PART FILE...'  
USE part  
PACK  
INDEX ON code TO part_cod  
INDEX ON weight TO part_wg  
?  
? 'PRODUCT FILE...'  
USE product  
PACK  
INDEX ON pd_code TO prod_cod  
? 'FORMULATION FILE...'
```

```
USE used
PACK
INDEX ON code TO used_cod
? 'STOCK INPUT FILE...'
USE stk_rec
PACK
INDEX ON date TO rec_date
? 'STOCK OUTPUT FILE...'
USE stk_shp
PACK
INDEX ON date TO shp_date
? 'STOCK BALANCE FILE...'
USE stk_bal
PACK
INDEX ON date TO bal_date
? 'PART PURCHASE FILE...'
USE partp
PACK
INDEX ON code TO partp_cod
? 'FORMULAR PURCHASE PART ...'
USE usep
PACK
INDEX ON code TO usep_cod
ENDCASE
CLOSE DATA
DO dbase
SET TALK OFF
DEACTIVATE WINDOW working
RETURN
```

```
*-----<BACKUP_DATA >----*

PROCEDURE backup

PARAMETER va1,va2,va3,va4,va5,va6

WAIT  'INSERT DISKETTE TO BACKUP FILE TO DRIVE ' + va1 + 'NOW !! '

SET DEFA TO &va1

IF DISKSPACE() > 1.1 * (va4 * 3 )

@ 5,0 SAY 'ENOUGH DISKSPACE TO BACKUP FILE '

CLOSE DATA

USE &va2

COPY TO  &va1&va3

@ 6,0 SAY  'BACKUP FILE COMPLETELY PRESS ANY KEY TO CONTINUE'

WAIT

CLOSE DATA

ELSE

@ 5,0 SAY 'NOT ENOUGH !! DISKSPACE TO BACKUP FILE'

ENDIF

CLOSE DATA

DEACTIVATE WINDOW back

SET DEFA TO &va5&va6

RETURN

*--< TRANS_PROC >--*

PROCEDURE trans_proc

DO CASE

CASE BAR () = 1

    editmode = 1

    ACTIVATE POPUP pop_edit

CASE BAR () = 2

    editmode = 2

    ACTIVATE POPUP pop_edit
```



```
CASE BAR () = 4
DO stk_in
CASE BAR () = 5
DO stk_out
ENDCASE
RETURN
*--< VIEW_PROC >--*
PROCEDURE view_proc
DEFINE WINDOW clr_scr FROM 0,0 TO 24,79 NONE
ACTIVATE WINDOW clr_scr
DO CASE
CASE BAR() = 3
    DO po_view
CASE BAR() = 4
    DO in_view
CASE BAR() = 5
    DO out_view
CASE BAR() = 6
    DO bal_view
CASE BAR() = 10
    DO product_view
CASE BAR() = 11
    DO part_view
CASE BAR() = 12
    DO pro_view
CASE BAR() = 13
    DO use_view
CASE BAR() = 14
    DO pur_view
```

```
CASE BAR() = 15
  DO usez_view
ENDCASE

RELEASE WINDOW clr_scr

RETURN

*--< INTER_PROC >--*

PROCEDURE inter_proc

DO CASE

CASE BAR() = 1

DO parta

CASE BAR() = 3
      DO product

CASE BAR() = 5
      DO process

CASE BAR() = 7

DO formular

      ENDCASE

RETURN

*--< PLAN_PROC >--*

PROCEDURE plan_proc

DO CASE

CASE BAR () = 1
      DO part_wan

CASE BAR () = 3
      DO po_wei

CASE BAR () = 11
      DO po_rep2

CASE BAR() = 12

DO pur_wei
```

```
ENDCASE
RETURN
*--< REPORT_PROC >--*
PROCEDURE report_proc
HIDE MENU main
HIDE POPUP pop_report
DO CASE
CASE BAR () = 2
SELECT pox
DO yesno_prn
IF pub_yn
DO printer
LIST FIELD job,pd_code,qty,date,remarkOFFTO PRINT
ELSE
LIST FIELD job,pd_code,qty,date,remarkOFF
WAIT
ENDIF
CASE BAR() = 3
SELECT partx
DO yesno_prn
IF pub_yn
DO printer
? CHR(15)
LIST FIELD code,name,material,size,drawing,turret,lvd,weight OFF TO PRINT
? CHR(18)
ELSE
LIST FIELD code,name,material,size,drawing,turret,lvd,weight OFF
WAIT
ENDIF
```

```
CASE BAR() = 4
SELECT productx
DO yesno_prn
IF pub_yn
DO printer
? CHR(15)
LIST FIELD pd_code,ue,cm,customer,applicat,system,packing,remar OFF TO PRINT
? CHR(18)
ELSE
LIST FIELD pd_code,ue,cm,customer,applicat,system,packing,remark OFF
WAIT
ENDIF
CASE BAR() = 7
SELECT partx
SELECT stk_rx
SET RELATION TO code INTO partx
DO yesno_prn
IF pub_yn
DO printer
LIST FIELD code,partx -> name,date,qty OFF TO PRINT
ELSE
LIST FIELD code,partx -> name,date,qty OFF
WAIT
ENDIF
CASE BAR() = 8
SELECT partx
SELECT stk_sx
SET RELATION TO code INTO partx
DO yesno_prn
```



```
IF pub_yn
DO printer
LIST FIELD code,partx -> name,date,qty OFF TO PRINT
ELSE
LIST FIELD code,partx -> name,date,qty OFF
WAIT
ENDIF

CASE BAR() = 9
SELECT partx
SELECT stk_bx
SET RELATION TO code INTO partx
DO yesno_prn
IF pub_yn
DO printer
LIST FIELD code,partx -> name,date,qty OFF TO PRINT
ELSE
LIST FIELD code,partx -> name,date,qty OFF
WAIT
ENDIF
ENDCASE

CLEAR

RETURN

*-< EXIT_PROC >--*

PROCEDURE exit_proc
DEFINE WINDOW exit_w FROM 8,16 TO 12,64
answer = "NO "
DO CASE
CASE BAR() = 1
ACTIVATE WINDOW exit_w
```

```
@ 1,7 SAY 'RETURN TO DOT PROMPT ';
GET answer FUNCTION "M NO,YES"
READ
pub_exit = IIF("NO" $ answer , .F. ,.T.)
IF pub_exit
CLOSE ALL
ENDIF
CASE BAR() = 2
ACTIVATE WINDOW exit_w
@ 1,10 SAY 'RETURN TO DOS ';
GET answer FUNCTION "M NO,YES"
READ
pub_exit = IIF("NO" $ answer ,.F. ,.T.)
IF pub_exit
    CLOSE ALL
QUIT
ENDIF
ENDCASE
RELEASE WINDOW exit_w
IF pub_exit
DEACTIVATE MENU
DEACTIVATE WINDOW ALL
CLOSE ALL
ENDIF
*-----< OPEN DBASE FILES TO SPECIFY PATH >-----*
PROCEDURE opendb
    PARAMETER filename
    PATH = SUBSTR(FILENAME,1,10)
    SET PATH TO (PATH)
```

DEACTIVATE POPUP

RETURN

--< EDIT_PROC >--

PROCEDURE edit_proc

DO CASE

CASE BAR () = 1 .AND. editmode = 1

DO po

CASE BAR () = 3 .AND. editmode = 1

DO po_up

CASE BAR () = 1 .AND. editmode = 2

DO balance

CASE BAR() = 3 .AND. editmode = 2

DO up_bal

ENDCASE

RETURN

----< PO_VIEW >----

PROCEDURE po_view

SELECT pox

head_msg = 'VIEW MANUFACTURING ORDER'

fieldname = 'field job,pd_code,qty,date,remark'

DO view_file WITH head_msg,fieldname,0,0,24,79

RETURN

---< IN_VIEW >---

PROCEDURE in_view

SELECT stk_rx

head_msg = 'VIEW PART INPUT TO STOCK'

fieldname = 'field code,date,qty'

DO view_file WITH head_msg,fieldname,0,0,24,79

RETURN

---<OUT_VIEW >---

PROCEDURE out_view

SELECT stk_sx

head_msg ='V I E W P A R T O U T P U T T O S T O C K '

fieldname ='field code,date,qty'

DO view_file WITH head_msg,fieldname,0,0,24,79

RETURN

---< BAL_VIEW >---

PROCEDURE bal_view

SELECT stk_bx

head_msg ='V I E W C U R R E N T S T O C K P A R T '

fieldname ='field code,date,qty'

DO view_file WITH head_msg,fieldname,0,0,24,79

RETURN

---< PRODUCT_VIEW>---

PROCEDURE product_view

SELECT productx

head_msg = 'V I E W P R O D U C T O F A I R '

fieldname = 'field pd_code,customer,ue,cm,color,applicat,fin,val,compress,;

heate,system,packing,kit,hilo,timer,thermo,transfer,socket,remark,option'

DO view_file WITH head_msg,fieldname,0,0,24,79

---< PART_VIEW >---

PROCEDURE part_view

SELECT partx

head_msg = 'V I E W P A R T T O P R O D U C T '

fieldname = 'field code,name,weight,material,size,;

drawing,turret,lvd,amada'

DO view_file WITH head_msg,fieldname,0,0,24,79

RETURN

---< PRO_VIEW >---

PROCEDURE pro_view

SELECT partx

head_msg = 'V I E W P R O C E S S T I M E (M I N)'

fieldname = 'field code,c_lvd,cs_lvd,c_10,cs_10,c_4,cs_4,p_ter,ps_ter,;

ps_sma,ps_med,ps_lar,p_rex,ps_rex,p_uni,ps_uni,p_sma,p_med,p_lar,;

b_lvd,bs_lvd,b_ama,bs_ama,b_kle,bs_kle,b_cha,bs_cha,b_24,bs_24,face,sw,arc,'

DO view_file WITH head_msg,fieldname,0,0,24,79

RETURN

---< USE_VIEW >---

PROCEDURE use_view

SELECT usex

head_msg = 'V I E W P A R T U S E P E R M O D E L '

fieldname = 'field code,pd_code,used,option,weight'

DO view_file WITH head_msg,fieldname,0,0,24,79

RETURN

---<PUR_VIEW>---

PROCEDURE pur_view

SELECT purx

head_msg = 'V I E W P U R C H A S E D P A R T '

fieldname = 'field code,name'

DO view_file WITH head_msg,fieldname,0,0,24,79

RETURN

---<USEZ_VIEW>---

PROCEDURE usez_view

SELECT usez

head_msg = 'V I E W F O R M U L A T I O N O F P U R . P A R T '

fieldname = 'field code,pd_code,used,option'

DO view_file WITH head_msg,fieldname,0,0,24,79

```
RETURN
*---< VIEW_FILE >---*
PROCEDURE view_file
PARAMETER head_msg,fieldname,srow,scol,frow,fcoll
IF reccount() >0
CLEAR
@ srow,scol TO srow + 2,fcoll
@ srow + 1 ,20 SAY head_msg
DEFINE WINDOW bd_w FROM srow + 3 ,scol TO frow,fcoll
ACTIVATE WINDOW bd_w
DEFINE WINDOW brow_w FROM srow + 4,scol+1 TO frow-1,fcoll-1 NONE
ACTIVATE WINDOW brow_w
GO TOP
browse &fieldname TITLE SPACE(10)
RELEASE WINDOW brow_w
RELEASE WINDOW bd_w
DO pack_file WITH DBF()
ENDIF
RETURN
*--< PACK_FILE >--*
PROCEDURE pack_file
PARAMETER filename
GO TOP
DO WHILE .NOT. EOF()
IF DELETE() = .T.
DO standby WITH 'PACK FILE .....PLEASE WAIT'
PACK
EXIT
ENDIF
```

```
SKIP
ENDDO
RETURN
*--< STANDBY >--*
PROCEDURE standby
PARAMETERS msgwords
DEFINE WINDOW standby FROM 19,12 TO 23,67
ACTIVATE WINDOW standby
@ 00,01 SAY msgwords
@ 01,01
WAIT ' <PRESS A KEY OR CLICK THE MOUSE TO CONTINUE> '
DEACTIVATE WINDOW standby
RETURN
*---< YESNO >---*
PROCEDURE yesno
PARAMETERS message
DEFINE WINDOW yn FROM 10,12 TO 14,67 DOUBLE FLOAT SHADOW COLOR SCHEME 7
ACTIVATE WINDOW yn
STORE 'NO' TO yn
? CHR(7)
@ 0,1 SAY 'BE MORE CAREFULLY !!'
@ 2,1 SAY 'DO YOU WANT TO CONTINUE ? NO/YES :' GET yn FUNCTION 'M NO,YES'
READ
pub_yn = IIF('NO' $ yn , .F. , .T. )
DEACTIVATE WINDOW yn
RETURN
*---< YESNOP >---*
PROCEDURE yesnop
PARAMETERS message
```

```

DEFINE WINDOW yn FROM 10,12 TO 14,67 DOUBLE FLOAT SHADOW COLOR SCHEME 7
ACTIVATE WINDOW yn
STORE 'NO' TO yn
? CHR(7)
@ 1,1 SAY 'DO YOU WANT TO SELECT SHEET METAL PART ? NO/YES : ' GET yn
READ
pub_yn = IIF('NO' $ yn , .F. , .T. )
DEACTIVATE WINDOW yn
RETURN
*---< YESNO_PRINT >---*
PROCEDURE yesno_prn
DEFINE WINDOW yn_prn FROM 8,12 TO 12,67 DOUBLE FLOAT SHADOW COLOR SCHEME 7
ACTIVATE WINDOW yn_prn
STORE 'NO' TO yn
? CHR(7)
@ 1,1 SAY 'DO YOU WANT TO BE PRINT ON PRINTER ? NO/YES : ' GET yn
READ
pub_yn = IIF('NO' $ yn , .F. , .T. )
DEACTIVATE WINDOW yn_prn
RETURN
*---< PRINTER >---*
PROCEDURE printer
STORE PRINTSTATUS() TO ready
IF .NOT. ready
DO standby WITH 'PRINTER IS NOT READY,PLEASE TURN ON PLEASE !!'
ELSE
DO standby WITH 'PRINTER IS READY'
ENDIF

```



```
pub_print =IIF(pub_print = "OFF","ON","OFF")
```

```
SET PRINTER &pub_print
```

```
RETURN
```

```
*----< CENTER >----*
```

```
FUNCTION center
```

```
PARAMETER mstring,mwidth
```

```
RETURN (mwidth-LEN(mstring))/2
```

```
*----<GETPART POPUP>----*
```

```
FUNCTION getpart
```

```
PARAMETER mname
```

```
SELECT partx
```

```
LOCATE FOR name = mname
```

```
IF .NOT. FOUND()
```

```
ACTIVATE SCREEN
```

```
IF EMPTY(PROMPT())
```

```
HIDE POPUP getpart
```

```
ELSE
```

```
SEEK PROMPT()
```

```
ENDIF
```

```
ENDIF
```

```
RETURN
```

```
*----<GETPO>----*
```

```
FUNCTION getpo
```

```
PARAMETER mpo
```

```
SELECT pox
```

```
LOCATE FOR job = mpo
```

```
IF .NOT. FOUND()
```

```
ACTIVATE SCREEN
```

```
IF EMPTY(PROMPT())
```

```

HIDE POPUP getpo
ELSE
SEEK PROMPT()
ENDIF
ENDIF
RETURN
*----<SET_CHOICE>----*
PROCEDURE set_choice
DEFINE MENU choice
DEFINE PAD search OF choice PROMPT '\<FIND' AT 23,2
DEFINE PAD next OF choice PROMPT '\<NEXT' AT 23,9
DEFINE PAD prev OF choice PROMPT '\<BEFOR' AT 23,16
DEFINE PAD first OF choice PROMPT '\<TOP' AT 23,23
DEFINE PAD last OF choice PROMPT '\<LAST' AT 23,28
DEFINE PAD edit OF choice PROMPT '\<EDIT' AT 23,35
DEFINE PAD append OF choice PROMPT '\<ADD' AT 23,41
DEFINE PAD delete OF choice PROMPT '\<DELETE' AT 23,46
DEFINE PAD print OF choice PROMPT '\<PRINT' AT 23,54
DEFINE PAD quit OF choice PROMPT '\<QUIT' AT 23,62
ON SELECTION PAD search OF choice DO srch_choice
ON SELECTION PAD next OF choice DO next_choice
ON SELECTION PAD prev OF choice DO prev_choice
ON SELECTION PAD first OF choice DO fir_choice
ON SELECTION PAD last OF choice DO las_choice
ON SELECTION PAD edit OF choice DO edit_choice
ON SELECTION PAD append OF choice DO app_choice
ON SELECTION PAD delete OF choice DO del_choice
ON SELECTION PAD print OF choice DO pri_choice
ON SELECTION PAD quit OF choice DO quit_choice

```

```
RETURN
*---<PO_UP>---*
PROCEDURE po_up
HIDE MENU main
HIDE POPUP pop_trans
HIDE POPUP pop_edit
@ 12,10 SAY ' PLEASE WAIT A MINUTE WHILE UPDATE PO.DBF'
SELECT tpox
GO TOP
DO WHILE .NOT. EOF()
IF DELETED()
SELECT pox
LOCATE FOR job = tpox -> job
DELETE FOR job = tpox -> job
ENDIF
SELECT tpox
SKIP
ENDDO
SELECT tpox
GO TOP
DO WHILE .NOT. EOF() .AND. .NOT. DELETED()
SELECT pox
LOCATE FOR job = tpox -> job
IF .NOT. FOUND()
APPEND BLANK
REPLACE job WITH tpox -> job
REPLACE pd_code WITH tpox -> pd_code
REPLACE qty WITH tpox -> qty
REPLACE date WITH tpox -> date
```

```
REPLACE remark WITH tpox -> remark
ENDIF
IF FOUND()
REPLACE pd_code WITH tpox -> pd_code
REPLACE qty WITH tpox -> qty
REPLACE date WITH tpox -> date
REPLACE remark WITH tpox -> remark
ENDIF
SELECT tpox
SKIP
ENDDO
SELECT tpox
PACK
REINDEX
SELECT pox
PACK
REINDEX
CLEAR
DO standby WITH ' UPDATE MASTER PO FILE ALREADY'
RETURN
*---<UP_BAL>---*
PROCEDURE up_bal
HIDE MENU main
HIDE POPUP pop_trans
HIDE POPUP pop_edit
@ 12,10 SAY ' PLEASE WAIT A MINUTE WHILE UPDATE STK_BAL.DBF'
SELECT stk_rx &&{UPDATE RECEIVE STOCK }
GO TOP
DO WHILE .NOT. EOF() .AND. .NOT. DELETED()
```



```
SELECT stk_bx
LOCATE FOR code = stk_rx -> code
IF .NOT. FOUND()
APPEND BLANK
REPLACE code WITH stk_rx -> code
REPLACE date WITH stk_rx -> date
REPLACE qty WITH stk_rx -> qty
ENDIF
IF FOUND()
REPLACE code WITH stk_rx -> code
REPLACE date WITH stk_rx -> date
REPLACE qty WITH (qty + (stk_rx -> qty))
ENDIF
SELECT stk_rx
SKIP
ENDDO
SELECT stk_sx &&{ UPDATE SHIP STOCK }
GO TOP
DO WHILE .NOT. EOF() .AND. .NOT. DELETED()
SELECT stk_bx
LOCATE FOR code = stk_sx -> code
IF .NOT. FOUND()
APPEND BLANK
REPLACE code WITH stk_sx -> code
REPLACE date WITH stk_sx -> date
REPLACE qty WITH stk_sx -> qty
ENDIF
IF FOUND()
REPLACE code WITH stk_sx -> code
```

```
REPLACE date WITH stk_sx -> date
REPLACE qty WITH (qty - (stk_sx -> qty))
ENDIF
SELECT stk_sx
SKIP
ENDDO
SELECT stk_rx
DELETE ALL
PACK
REINDEX
SELECT stk_sx
DELETE ALL
PACK
REINDEX
SELECT stk_bx
PACK
REINDEX
CLEAR
DO standby WITH ' UPDATE INPUT PART ON MASTER STK_BAL FILE ALREADY'
RETURN
*EOF
```

```

*****
*PROGRAM : PO_WEI.PRG
*THIS PROGRAM USE
*(1) TO FIND ORDER USE PER PART = ?
*(2) TO FIND M/C HRS USE OF EACH STATION
*****

HIDE MENU main

HIDE POPUP pop_plan

STORE .F. TO go_new

STORE .F. TO go_prn

STORE .T. TO logic

DO setup_pox

DO disp_pox

STORE .T. TO going

DO WHILE going

ACTIVATE WINDOW pox_wind

ACTIVATE MENU pox PAD pnew

ENDDO

DO shut_pox

CLEAR

*---< SETUP_POX >----*

PROCEDURE setup_pox

ACTIVATE SCREEN

CLEAR

DEFINE WINDOW pox_wind FROM 0,0 TO 4,79 ;

TITLE 'PLAN BY SELECT PO JOB 'SYSTEM GROW ZOOM COLOR SCHEME 10

DEFINE WINDOW pa_wind FROM 4,0 TO 22,79 ;

TITLE 'CODE TOTAL JOB 1JOB 2JOB 3JOB 4JOB 5JOB 6JOB 7JOB 8JOB 9JOB10';

SYSTEM GROW ZOOM COLOR SCHEME 10

```

```
ACTIVATE SCREEN
IF go_prn = .F.
IF go_new = .F.
DEFINE MENU pox COLOR SCHEME 3
DEFINE PAD pnew OF pox PROMPT '\<NEW ' AT 23,1
DEFINE PAD pquit OF pox PROMPT '\<QUIT' AT 23,9
ENDIF
ENDIF
ON SELECTION PAD pnew OF pox DO pnew_pox
ON SELECTION PAD pquit OF pox DO quit_pox
RETURN
*---<DISP_POX>---*
PROCEDURE disp_pox
ans = 0
ans2 = SPACE(6)
a = 1
b = 1
ACTIVATE WINDOW pox_wind
DO WHILE logic
@ 0,0 SAY 'HOW MANY JOBS TO SELECT (1..8)'
@ 0,32 GET ans
@ 0,45 SAY 'INDEX BY'
@ 0,55 GET ans2 FUNCTION '*R' PICTURE 'CODE;PRIORITY'
READ
IF ans <= 0 .OR. ans >=9
DO standby WITH 'INVALID NO. JOB PLEASE ENTER AGAIN'
LOOP
ELSE
EXIT
```



```
ENDIF
ENDDO
DECLARE mjob(ans,12)      && {collect job selected in array}
DO WHILE a <= ans
SELECT pox
ACTIVATE POPUP getpo
STORE job      TO mjob(a,1)
STORE pd_code TO mjob(a,2)
STORE qty      TO mjob(a,3)
STORE date     TO mjob(a,4)
STORE SUBSTR(pd_code,1,4) TO po1
STORE SUBSTR(pd_code,6,2) TO po2
SELECT productx
GO TOP
DO WHILE .NOT. EOF()
STORE SUBSTR(pd_code,1,4) TO product1
STORE SUBSTR(pd_code,6,2) TO product2
IF po1 = product1 .AND. po2 = product2
STORE color      TO mjob(a,5)
STORE customer TO mjob(a,6)
STORE applicat TO mjob(a,7)
STORE val        TO mjob(a,8)
STORE system     TO mjob(a,9)
STORE packing    TO mjob(a,10)
STORE thermo     TO mjob(a,11)
EXIT
ENDIF
SKIP
ENDDO
```

```
IF a >= 8
@ 2,(11*b)-11 SAY ALLTRIM(mjob(a,1))
b = b + 1
ELSE
@ 1,(11*a)-11 SAY ALLTRIM(mjob(a,1))
ENDIF
a = a + 1
SELECT pox
ENDDO
SELECT partx
COUNT TO mcount
IF ans2 = 'CODE'
USE
SELECT 1
USE part INDEX part_cod ALIAS partx
SELECT partx
ENDIF
IF ans2 = 'PRIORITY'
USE
SELECT 1
USE part INDEX part_wg ALIAS partx
SELECT partx
ENDIF
*-----BEGIN TO PA_WIND & COLLECT ALL PO -----*
ACTIVATE WINDOW pa_wind
DO blk_moni
@ 6,15 TO 10,68
@ 8,20 SAY 'PLEASE WAIT A MINUTE WHILE CALCULATE !'
DO moniter
```

```
col = 1
co2 = 1
co3 = 1
co4 = 1
co5 = 1
co6 = 1
co7 = 1
co8 = 1

IF ans >= 1
IF ans >= 2
IF ans >= 3
IF ans >= 4
IF ans >= 5
IF ans >= 6
IF ans >= 7
IF ans >= 8

DECLARE t_po8(70,3)

DO coll_dataWITH mjob(8,2),mjob(8,3),t_po8,co8

ENDIF && ans= 8

DECLARE t_po7(70,3)

DO coll_dataWITH mjob(7,2),mjob(7,3),t_po7,co7

ENDIF&& ans = 7

DECLARE t_po6(70,3)

DO coll_dataWITH mjob(6,2),mjob(6,3),t_po6,co6

ENDIF&& ans =6

DECLARE t_po5(70,3)

DO coll_dataWITH mjob(5,2),mjob(5,3),t_po5,co5

ENDIF && ans = 5

DECLARE t_po4(70,3)
```

```
DO coll_data WITH mjob(4,2),mjob(4,3),t_po4,co4
ENDIF && ans = 4
DECLARE t_po3(70,3)
DO coll_data WITH mjob(3,2),mjob(3,3),t_po3,co3
ENDIF && ans =3
DECLARE t_po2(70,3)
DO coll_data WITH mjob(2,2),mjob(2,3),t_po2,co2
ENDIF && ans = 2
DECLARE t_po1(70,3)
DO coll_data WITH mjob(1,2),mjob(1,3),t_po1,co1
ENDIF && ans = 1
CLEAR
*-----BEGIN TO FIND TOTAL EACH PO -----*
x      = 1
rw1    = 1
rw2    = 1
rw3    = 1
rw4    = 1
rw5    = 1
rw6    = 1
rw7    = 1
rw8    = 1
DECLARE sum(mcount,2)
DECLARE temp1(145,2)
DECLARE temp2(145,2)
DECLARE temp3(145,2)
DECLARE temp4(145,2)
DECLARE temp5(145,2)
DECLARE temp6(145,2)
```



```
DECLARE temp7(145,2)
DECLARE temp8(145,2)
DO set_prn  && PRINT OR NOT ?
IF go_prn = .T.
said = 'PLAN TOTAL PART TO FABRICATE ON '
DO title WITH said
line = 17
an1  = 42
an2  = 47
an3  = 5
ELSE
line = 0
an1  = 0
an2  = 0
an3  = 0
ENDIF&& go_prn
SELECT partx
GO TOP
DO WHILE .NOT. EOF()
sumpo = 0
meet1 = ASCAN(t_pol,code)  && {partx -> code = ARRAY -> code}
IF EMPTY(meet1) # .T.
rw1 = ASUBSCRIPT(t_pol,meet1,1)
@ line,18 + an1 SAY ALLTRIM(STR(t_pol(rw1,2)))
sumpo = sumpo + t_pol(rw1,2)
STORE t_pol(rw1,1) TO sum(x,1)
STORE t_pol(rw1,2) TO temp1(x,1)
STORE 1 TO temp1(x,2)
ELSE
```

```
STORE 0 TO temp1(x,2)
ENDIF
IF ans >= 2
meet2 = ASCAN(t_po2,code)
IF EMPTY(meet2) # .T.
rw2 = ASUBSCRIPT(t_po2,meet2,1)
@ line,24 + an2 SAY ALLTRIM(STR(t_po2(rw2,2)))
sumpo = sumpo + t_po2(rw2,2)
STORE t_po2(rw2,1) TO sum(x,1)
STORE t_po2(rw2,2) TO temp2(x,1)
STORE 1 TO temp2(x,2)
ELSE
STORE 0 TO temp2(x,2)
ENDIF
IF ans >= 3
meet3 = ASCAN(t_po3,code)
IF EMPTY(meet3) # .T.
rw3 = ASUBSCRIPT(t_po3,meet3,1)
@ line,30 + an2 + an3 SAY ALLTRIM(STR(t_po3(rw3,2)))
sumpo = sumpo + t_po3(rw3,2)
STORE t_po3(rw3,1) TO sum(x,1)
STORE t_po3(rw3,2) TO temp3(x,1)
STORE 1 TO temp3(x,2)
ELSE
STORE 0 TO temp3(x,2)
ENDIF
IF ans >= 4
meet4 = ASCAN(t_po4,code)
IF EMPTY(meet4) # .T.
```

```
rw4 = ASUBSCRIPT(t_po4,meet4,1)
@ line,36 + an2 + (2 * an3) SAY ALLTRIM(STR(t_po4(rw4,2)))
sumpo = sumpo + t_po4(rw4,2)
STORE t_po4(rw4,1) TO sum(x,1)
STORE t_po4(rw4,2) TO temp4(x,1)
STORE 1 TO temp4(x,2)
ELSE
STORE 0 TO temp4(x,2)
ENDIF
IF ans >= 5
meet5 = ASCAN(t_po5,code)
IF EMPTY(meet5) # .T.
rw5 = ASUBSCRIPT(t_po5,meet5,1)
@ line,42 + an2 + (3 * an3) SAY ALLTRIM(STR(t_po5(rw5,2)))
sumpo = sumpo + t_po5(rw5,2)
STORE t_po5(rw5,1) TO sum(x,1)
STORE t_po5(rw5,2) TO temp5(x,1)
STORE 1 TO temp5(x,2)
ELSE
STORE 0 TO temp5(x,2)
ENDIF
IF ans >= 6
meet6 = ASCAN(t_po6,code)
IF EMPTY(meet6) # .T.
rw6 = ASUBSCRIPT(t_po6,meet6,1)
@ line,47 + an2 + (4 * an3) SAY ALLTRIM(STR(t_po6(rw6,2)))
sumpo = sumpo + t_po6(rw6,2)
STORE t_po6(rw6,1) TO sum(x,1)
STORE t_po6(rw6,2) TO temp6(x,1)
```

```
STORE 1 TO temp6(x,2)
ELSE
STORE 0 TO temp6(x,2)
ENDIF

IF ans >= 7
meet7 = ASCAN(t_po7,code)
IF EMPTY(meet7) # .T.
rw7 = ASUBSCRIPT(t_po7,meet7,1)
@ line,53 + an2 + (5 * an3) SAY ALLTRIM(STR(t_po7(rw7,2)))
sumpo = sumpo + t_po7(rw7,2)
STORE t_po7(rw7,1) TO sum(x,1)
STORE t_po7(rw7,2) TO temp7(x,1)
STORE 1 TO temp7(x,2)
ELSE
STORE 0 TO temp7(x,2)
ENDIF

IF ans >= 8
meet8 = ASCAN(t_po8,code)
IF EMPTY(meet8) # .T.
rw8 = ASUBSCRIPT(t_po8,meet8,1)
@ line,60 + an2 + (6 * an3) SAY ALLTRIM(STR(t_po8(rw8,2)))
sumpo = sumpo + t_po8(rw8,2)
STORE t_po8(rw8,1) TO sum(x,1)
STORE t_po8(rw8,2) TO temp8(x,1)
STORE 1 TO temp8(x,2)
ELSE
STORE 0 TO temp8(x,2)
ENDIF

ENDIF && ans >= 8
```



```
ENDIF && ans >= 7
ENDIF && ans >= 6
ENDIF && ans >= 5
ENDIF && ans >= 4
ENDIF && ans >= 3
ENDIF && ans >= 2
STORE sumpo TO sum(x,2)
IF sum(x,2) # 0
IF go_prn = .T.
@ line,50 SAY ALLTRIM(STR(sum(x,2)))
ELSE
@ line,10 SAY ALLTRIM(STR(sum(x,2)))
ENDIF
ENDIF
IF EMPTY(temp1(x,2))# .T. .OR. EMPTY(temp2(x,2))# .T. .OR. ;
+ EMPTY(temp4(x,2))# .T. .OR. EMPTY(temp5(x,2))# .T. .OR. ;
+ EMPTY(temp7(x,2))# .T. .OR. EMPTY(temp8(x,2))# .T.
IF go_prn = .T.
@ line,0 SAY code
@ line,8 SAY name
ELSE
@ line,0 SAY code
ENDIF
x = x + 1 && {if place this because if not have code equal x not increase}
line = line + 1
IF go_prn = .F.
IF line = 14
WAIT
CLEAR
```

```
line = 0
ENDIF
ENDIF && go_prn
ELSE
ENDIF && check part code partx = part code po
stop_1 = x - 1    && {because reduce 1 step of x = x + 1 }
SKIP
ENDDO && {select partx}
RELEASE ALL LIKE t_po*
RELEASE ALL LIKE rw*
RELEASE ALL LIKE co*
@ line,0 SAY REPLICATE('-',76)
IF go_prn = .T.
EJECT
ENDIF
WAIT
SET DEVICE TO SCREEN
STORE .F. TO go_prn
*---MODULE BALANCE--->
RELEASE WINDOW pa_wind
DEFINE WINDOW pa_wind FROM 4,0 TO 22,79 ;
TITLE 'CODE    JOB    DUE DATE          QTY    IN HAND    BALANCE    PRODUCE ';
SYSTEM GROW ZOOM COLOR SCHEME 10
ACTIVATE WINDOW pa_wind
CLEAR
@ 5,10 SAY 'NEXT ! TO ALLOCATE PART TO ASSEMBLY PART STOCK WILL BE CHANGE'
WAIT
CLEAR
DECLARE tot_pd(mcount)
```

```
DO yesno
IF pub_yn
DO set_prn
x      = 1
line   = 0
DO WHILE x <= stop_1
IF EMPTY(sum(x,1)) # .T.  && {code not equal empty}
LOCATE FOR code = sum(x,1)
sumrec = 0
IF go_prn = .T.
@ line,0 SAY name
line = line + 1
ELSE
@ line,0 SAY sum(x,1)
ENDIF
IF EMPTY(temp1(x,2)) # .T.
DO reduce WITH temp1(x,2),mjob(1,1),mjob(1,4),temp1(x,1)
ENDIF
IF EMPTY(temp2(x,2)) # .T.
DO reduce WITH temp2(x,2),mjob(2,1),mjob(2,4),temp2(x,1)
ENDIF
IF EMPTY(temp3(x,2)) # .T.
DO reduce WITH temp3(x,2),mjob(3,1),mjob(3,4),temp3(x,1)
ENDIF
IF EMPTY(temp4(x,2)) # .T.
DO reduce WITH temp4(x,2),mjob(4,1),mjob(4,4),temp4(x,1)
ENDIF
IF EMPTY(temp5(x,2)) # .T.
DO reduce WITH temp5(x,2),mjob(5,1),mjob(5,4),temp5(x,1)
```

```
ENDIF
IF EMPTY(temp6(x,2)) # .T.
  DO reduce WITH temp6(x,2),mjob(6,1),mjob(6,4),temp6(x,1)
ENDIF
IF EMPTY(temp7(x,2)) # .T.
  DO reduce WITH temp7(x,2),mjob(7,1),mjob(7,4),temp7(x,1)
ENDIF
IF EMPTY(temp8(x,2)) # .T.
  DO reduce WITH temp8(x,2),mjob(8,1),mjob(8,4),temp8(x,1)
ENDIF
IF go_prn = .T.
  line = line + 1
ENDIF
@ line,21 SAY 'TOTAL ='
@ line,30 SAY sum(x,2)
SELECT stk_bx
SUM qty FOR code = sum(x,1) TO sumrec
STORE sum(x,1) TO tot_pd(x)
@ line,40 SAY sumrec
tot_pd(x) = (sumrec - sum(x,2))
@ line,50 SAY tot_pd(x)
IF tot_pd(x) < 0
@ line,65 SAY ABS(tot_pd(x))
ELSE
@ line,74 SAY 0
ENDIF
line = line + 1
IF go_prn = .F.
IF line >= 7
```



```

WAIT
CLEAR
line = 0
ENDIF
ENDIF
ENDIF && empty(sum(x,1) = .T.
x = x + 1
ENDDO
ELSE
RETURN
ENDIF && pub_yn
RELEASE ALL LIKE temp*
SET DEVICE TO SCREEN
STORE .F. TO go_prn
*-----BEGIN PLAN CUT MACHINE -----*
RELEASE WINDOW pa_wind
DEFINE WINDOW pa_wind FROM 4,0 TO 22,79 ;
TITLE 'ID CODE PD tLVD t10 t4 CUM 1 CUM2 CUM3 M/C TIME CUM.T';
SYSTEM GROW ZOOM COLOR SCHEME 10
ACTIVATE WINDOW pa_wind
c_hr = 0.0 && {collect working hour}
c_mc = 0 && {collect machine hour}
c_mn = 0 && {collect man hour}
cb_min = 0.0 && { time machine stop work }
a_prn = 0 && {add col to print on printer}
CLEAR
@ 3,10 SAY 'BEGIN TO PLAN JOB IN CUTTING MACHINE'
@ 4,10 SAY REPLICATE('=',46)
@ 5,10 SAY 'HOW MANY HOURS DO YOU HAVE ?' GET c_hr

```

```

@ 5,60 SAY 'HOURS'

@ 6,10 SAY 'HOW MANY MACHINE DO YOU WORK ?' GET c_mc

@ 7,10 SAY 'HOW MANY MEN      DO YOU WORK ?' GET c_mn

READ

cb_min = ( c_hr * 60 ) * c_mc

WAIT

CLEAR

DO yesno

IF pub_yn

DO set_prn

IF go_prn = .T.

@ 0,5 SAY ' J O B   P L A N N I N G   O F   C U T T I N G   S T A T I O N   '

@ 2,0 SAY "ID CODE      NAME                PRODUCE MATE. SIZE M/C M/C MAN"

@ 3,0 SAY "                                (PCS)          (mm*mm) NO. MIN MIN"

@ 4,0 SAY REPLICATE('-',76)

line = 5

a_prn = 30

ELSE

line = 0

a_prn = 0

ENDIF

SELECT partx.

GO TOP

x      = 1

z      = 1

over_c = 0

accum1 = 0

accum2 = 0

accum3 = 0

```

```
accumCM = 0
man_c   = 0.0
c_set1  = 0
c_set2  = 0
c_set3  = 0
to_cum1 = 0
to_cum2 = 0
to_cum3 = 0
check1  = .F.
check2  = .F.
check3  = .F.
toto    = 0
cum_cut = 0

DECLARE sort_part(stop_1,3)
DO WHILE x <= stop_1
accumCM = 0
IF EMPTY(sum(x,1)) # .T.
LOCATE FOR code = sum(x,1)  && {code part.dbf = code sum(x,1)}
@ line,0  SAY ALLTRIM(STR(x))
IF go_prn = .F.
@ line,3  SAY ALLTRIM(sum(x,1)) && {code}
ELSE
    @ line,3  SAY ALLTRIM(sum(x,1))
@ line,11 SAY name
ENDIF
IF tot_pd(x) <= 0
@ line,11 SAY ALLTRIM(STR(ABS(tot_pd(x))))
ELSE
@ line,11 SAY 0
```

```
ENDIF
IF go_prn = .T.
@ line,56 SAY material
@ line,62 SAY size
ENDIF
DO CASE
CASE c_lvd # 0 .AND. c_10 # 0 .AND. c_4 # 0
IF tot_pd(x) < 0
accum1 = (ABS(tot_pd(x)) * c_lvd) + cs_lvd
accum2 = (ABS(tot_pd(x)) * c_10 ) + cs_10
accum3 = (ABS(tot_pd(x)) * c_4 ) + cs_4
ELSE
accum1 = 0
accum2 = 0
accum3 = 0
ENDIF
DO CASE
CASE to_cum1 <= to_cum2 .AND. to_cum1 <= to_cum3
to_cum1 = to_cum1 + accum1
c_set1 = c_set1 + cs_lvd
accumCM = accumCM + (m_cut * accum1)
IF go_prn = .F.
@ line,19 SAY LTRIM(STR(accum1))
@ line,31 SAY to_cum1
ENDIF
check1 = .T.
CASE to_cum2 <= to_cum1 .AND. to_cum2 <= to_cum3
to_cum2 = to_cum2 + accum2
c_set2 = c_set2 + cs_10
```



```

accumCM = accumCM + (m_cut * accum2)
IF go_prn = .F.
@ line,25 SAY LTRIM(STR(accum2))
@ line,40 SAY to_cum2
ENDIF
check2 = .T.
CASE to_cum3 <= to_cum1 .AND. to_cum3 <= to_cum2
to_cum3 = to_cum3 + accum3
c_set3 = c_set3 + cs_4
accumCM = accumCM + (m_cut * accum3)
IF go_prn = .F.
@ line,31 SAY LTRIM(STR(accum3))
@ line,49 SAY to_cum3
ENDIF
check3 = .T.
ENDCASE
CASE c_lvd # 0 .AND. c_10 # 0 .AND. c_4 = 0
IF tot_pd(x) < 0
accum1 = (ABS(tot_pd(x)) * c_lvd) + cs_lvd
accum2 = (ABS(tot_pd(x)) * c_10 ) + cs_10
ELSE
accum1 = 0
accum2 = 0
ENDIF
DO CASE
CASE to_cum1 <= to_cum2
to_cum1 = to_cum1 + accum1
c_set1 = c_set1 + cs_lvd
accumCM = accumCM + ( m_cut * accum1)

```

```
IF go_prn = .F.
@ line,19 SAY LTRIM(STR(accum1))
@ line,31 SAY to_cum1
ENDIF

check1 = .T.

CASE to_cum2 <= to_cum1
to_cum2 = to_cum2 + accum2
c_set2 = c_set2 + cs_10
accumCM = accumCM + (m_cut * accum2)
IF go_prn = .F.
@ line,25 SAY LTRIM(STR(accum2))
@ line,40 SAY to_cum2
ENDIF

check2 = .T.

ENDCASE

CASE c_lvd # 0 .AND. c_10 = 0 .AND. c_4 # 0
IF tot_pd(x) < 0
accum1 = (ABS(tot_pd(x)) * c_lvd) + cs_lvd
accum3 = (ABS(tot_pd(x)) * c_4 ) + cs_4
ELSE
accum1 = 0
accum3 = 0
ENDIF

DO CASE
CASE to_cum1 <= to_cum3
to_cum1 = to_cum1 + accum1
c_set1 = c_set1 + cs_lvd
accumCM = accumCM + ( m_cut * accum1)
IF go_prn = .F.
```

```
@ line,19 SAY LTRIM(STR(accum1))
@ line,31 SAY to_cum1
ENDIF
check1 = .T.
CASE to_cum3 <= to_cum1
to_cum3 = to_cum3 + accum3
c_set3 = c_set3 + cs_4
accumCM = accumCM + (m_cut * accum3)
IF go_prn = .F.
@ line,31 SAY LTRIM(STR(accum3))
@ line,49 SAY to_cum3
ENDIF
check3 = .T.
ENDCASE
CASE c_lvd = 0 .AND. c_10 = 0 .AND. c_4 = 0
IF tot_pd(x) < 0
accum1 = 0
accum2 = 0
accum3 = 0
ENDIF
accumCM = 0
IF go_prn = .F.
@ line,19 SAY LTRIM(STR(accum1))
@ line,31 SAY to_cum1
ENDIF
check1 = .T.
ENDCASE&& select group m/c work (3 group)
DO CASE
CASE check1 = .T. && { run on LVD machine }
```

```
STORE sum(x,1) TO sort_part(x,1)  && collect code
STORE tot_pd(x) TO sort_part(x,2)  && collect qty
STORE to_cum1 TO sort_part(x,3)  && collect time
@ line,70 SAY LTRIM(STR(1))
@ line,73 SAY LTRIM(STR(accum1))
DO prn_data WITH line,76,accumCM
over_c = over_c + accum1
check1 = .F.
CASE check2 = .T.  && {run on 10 FT machine }
STORE sum(x,1) TO sort_part(x,1)
STORE tot_pd(x) TO sort_part(x,2)
STORE to_cum2 TO sort_part(x,3)
@ line,70 SAY LTRIM(STR(2))
@ line,73 SAY LTRIM(STR(accum2))
DO prn_data WITH line,76,accumCM
over_c = over_c + accum2
check2 = .F.
CASE check3 = .T.&& {run on 4 FT machine}
STORE sum(x,1) TO sort_part(x,1)
STORE tot_pd(x) TO sort_part(x,2)
STORE to_cum3 TO sort_part(x,3)
@ line,70 SAY LTRIM(STR(3))
@ line,73 SAY LTRIM(STR(accum3))
DO prn_data WITH line,76,accumCM
over_c = over_c + accum3
check3 = .F.
ENDCASE
toto = toto + sort_part(x,3)
DO say_data WITH line,76,accumCM
```



```
line = line + 1
IF go_prN = .F.
  IF line >= 13
    WAIT
    CLEAR
    line = 0
  ENDIF
ENDIF && go_prN = .F.
ENDIF && EMPTY(sum(x,1)) # .T.
x = x + 1 && {place after if because out if increase 1}
IF over_c >= (cb_min * z)
  @ line,0 SAY REPLICATE('-',76)
  line = line + 1
  z = z + 1
ENDIF
cum_cut = cum_cut + accumCM
ENDDO && stop_1
man_c = (cum_cut / 60) / c_hr
mc_hr = 0.00
mc_hr = over_c / 60
man_hr = cum_cut / 60
nam1 = 'LVD'
nam2 = '10 FT'
nam3 = '4 FT'
mes = 'PRESS ANY KEY TO CONCLUDE RESULT OF CUT STATION'
DO include WITH mes,c_hr,c_mc,c_mn,mc_hr,man_hr,man_c,to_cum1,c_set1
ELSE
RETURN
ENDIF && pub_yn
```

```
IF go_prn = .T.
EJECT
ENDIF

RELEASE ALL LIKE accum
RELEASE ALL LIKE to_cum
RELEASE ALL LIKE sum
SET DEVICE TO SCREEN
STORE .F. TO go_prn
WAIT

*-----BEGIN PLAN PUNCH MACHINE -----*

RELEASE WINDOW pa_wind
DEFINE WINDOW pa_wind FROM 4,0 TO 22,79 ;
TITLE 'ID CODE   PD  tT  tS  tM  tL  tU  tR  cuT  cuS  cuM  cuL  cuU';
SYSTEM GROW ZOOM COLOR SCHEME 10
ACTIVATE WINDOW pa_wind

p_hr   = 0.0
p_mc   = 0
p_man  = 0
pb_min = 0.0
over_p = 0
p_set1 = 0
p_set2 = 0
p_set3 = 0
p_set4 = 0

CLEAR

@ 3,10 SAY 'BEGIN TO PLAN JOB IN PUNCH MACHINE LINE'
@ 4,10 SAY REPLICATE('=',51)
@ 5,10 SAY 'HOW MANY HOURS DO YOU HAVE   ?' GET p_hr
@ 5,60 SAY 'HOURS'
```

```
@ 6,10 SAY 'HOW MANY MACHINE DO YOU WORK ?' GET p_mc
@ 7,10 SAY 'HOW MANY MEN      DO YOU WORK ?' GET p_man

READ

pb_min = ( p_hr * 60 ) * p_mc

WAIT

CLEAR

DO yesno

IF pub_yn

DO set_prn

IF go_prn = .T.

@ 0,5 SAY 'JOB PLANNING OF PUNCH STATION'
@ 2,0 SAY "ID NAME                PRODUCE DWG PRG  M/C  M/C  MAN"
@ 3,0 SAY "                        (PCS) 04647 TUR      (MIN)(MIN)"
@ 4,0 SAY REPLICATE('-',76)

line = 5

ELSE

line = 0

ENDIF

SELECT partx

GO TOP

DECLARE sor_pun(stop_1,3)

x      = 1

z      = 1

over_p = 0

accumT = 0

accumS = 0

accumM = 0

accumL = 0

accumU = 0
```

```
accumR = 0
accumPM = 0 && {collect PUNCH MAN POWER}
to_cumT = 0
to_cumS = 0
to_cumM = 0
to_cumL = 0
to_cumU = 0
to_cumR = 0
pick1 = 0
pick2 = 0
pick3 = 0
pick4 = 0
pick5 = 0
pick6 = 0
p_set1 = 0
p_set2 = 0
p_set3 = 0
p_set4 = 0
p_set5 = 0
p_set6 = 0
counter = 1
=ASORT(sort_part,3)
DO WHILE x <= stop_1
sum_cum = 0
man_cum = 0
IF EMPTY(sort_part(x,1)) # .T.
LOCATE FOR code = sort_part(x,1)
IF sort_part(x,2) < 0
accumT = (ABS(sort_part(x,2)) * p_ter) + ps_ter
```



```

accumS = (ABS(sort_part(x,2)) * p_sma) + ps_sma
accumM = (ABS(sort_part(x,2)) * p_med) + ps_med
accumL = (ABS(sort_part(x,2)) * p_lar) + ps_lar
accumU = (ABS(sort_part(x,2)) * p_uni) + ps_uni
accumR = (ABS(sort_part(x,2)) * p_rex) + ps_rex
p_set1 = p_set1 + ps_ter
p_set2 = p_set2 + ps_sma
p_set3 = p_set3 + ps_med
p_set4 = p_set4 + ps_lar
p_set5 = p_set5 + ps_uni
p_set6 = p_set6 + ps_rex
ELSE
accumT = 0
accumS = 0
accumM = 0
accumL = 0
accumU = 0
accumR = 0
ENDIF
over_p = over_p + (accumT + accumS + accumM + accumL + accumU + accumR )
accumPM = (accumPM + (accumT + accumS + accumM + accumL + accumU ) * m_pun)
IF accumT # 0 .OR. accumS # 0 .OR. accumM # 0 .OR. accumL # 0
@ line,0 SAY ALLTRIM(STR(counter))
counter = counter + 1
IF go_prn = .F.
@ line,3 SAY ALLTRIM(sort_part(x,1))
a_prn = 0
ELSE
@ line,3 SAY NAME

```

```
a_prn = 30
ENDIF
IF sort_part(x,2) <= 0
@ line ,13 + a_prn SAY ALLTRIM(STR(ABS(sort_part(x,2))))
ELSE
@ line ,13 + a_prn SAY 0
ENDIF
IF go_prn = .T.
@ line,49 SAY ALLTRIM(SUBSTR(drawing,6,3))
@ line,54 SAY ALLTRIM(turret)
ENDIF
IF go_prn = .F.
@ line,17 SAY LTRIM(STR(accumT))
@ line,21 SAY LTRIM(STR(accumS))
@ line,25 SAY LTRIM(STR(accumM))
@ line,29 SAY LTRIM(STR(accumL))
@ line,33 SAY LTRIM(STR(accumU))
@ line,37 SAY LTRIM(STR(accumR))
ENDIF
to_cumT = to_cumT + accumT
to_cumS = to_cumS + accumS
to_cumM = to_cumM + accumM
to_cumL = to_cumL + accumL
to_cumU = to_cumU + accumU
to_cumR = to_cumR + accumR
IF accumT # 0
DO say_data WITH line,41,to_cumT
DO prnldata WITH line,59,'T'
pick1 = to_cumT
```

```
ELSE
@ line,41 SAY ' '
pick1 = 0
ENDIF
IF accumS # 0
DO say_data WITH line,46,to_cumS
DO prnldata WITH line,60,'S'
pick2 = to_cumS
ELSE
@ line,46 SAY ' '
pick2 = 0
ENDIF
IF accumM # 0
DO say_data WITH line,51,to_cumM
DO prnldata WITH line,61,'M'
pick3 = to_cumM
ELSE
@ line,51 SAY ' '
pick3 = 0
ENDIF
IF accumL # 0
DO say_data WITH line,57,to_cumL
DO prnldata WITH line,62,'L'
pick4 = to_cumL
ELSE
@ line,57 SAY ' '
pick4 = 0
ENDIF
IF accumU # 0
```

```
DO say_data WITH line,62,to_cumU
DO prnldata WITH line,63,'U'
pick5 = to_cumU
ELSE
@ line,62 SAY ' '
pick5 = 0
ENDIF
IF accumR # 0
DO say_data WITH line,67,to_cumR
DO prnldata WITH line,64,'R'
pick6 = to_cumR
ELSE
@ line,67 SAY ' '
pick6 = 0
ENDIF
sum_cum = (accumT + accumS + accumM + accumL + accumU + accumR )
man_cum = sum_cum * m_pun
DO CASE
CASE pick1 >= pick2 .AND. pick1 >= pick3 .AND. pick1 >= pick4
STORE pick1 TO sort_part(x,3)
DO prn_data WITH line,67,sum_cum
CASE pick2 >= pick1 .AND. pick2 >= pick3 .AND. pick2 >= pick4
STORE pick2 TO sort_part(x,3)
DO prn_data WITH line,67,sum_cum
CASE pick3 >= pick1 .AND. pick3 >= pick2 .AND. pick3 >= pick4
STORE pick3 TO sort_part(x,3)
DO prn_data WITH line,67,sum_cum
CASE pick4 >= pick1 .AND. pick4 >= pick2 .AND. pick4 >= pick3
STORE pick4 TO sort_part(x,3)
```



```
DO prn_data WITH line,67,sum_cum
CASE pick5 >= pick1 .AND. pick5 >= pick2 .AND. pick5 >= pick3
STORE pick5 TO sort_part(x,3)
DO prn_data WITH line,67,sum_cum
CASE pick6 >= pick1 .AND. pick6 >= pick2 .AND. pick6 >= pick3
STORE pick6 TO sort_part(x,3)
DO prn_data WITH line,67,sum_cum
ENDCASE
@ LINE,72 SAY LTRIM(STR(man_cum))
line = line + 1
IF go_prn = .F.
IF line .>= 14
WAIT
CLEAR
line = 0
ENDIF
ENDIF
ENDIF && {accumT .. accumR = 0}
ENDIF && {sort_part(x,1) # .T.
x = x + 1
IF over_p >= (pb_min * z)
@ line,0 SAY REPLICATE('-',76)
line = line + 1
z = z + 1
ENDIF
ENDDO && stop_1
man_p = (accumPM / 60 ) /p_hr
mc_hr = 0.00
mc_hr = (over_p / 60)
```

```
man_hr = accumPM / 60
nam1   = 'TUR'
nam2   = 'SMAL.'
nam3   = 'MEDI.'
nam4   = 'LARG.'
nam5   = 'UNIP.'
nam6   = 'REXR.'
mes    = 'PRESS ANY KEY TO CONCLUDE RESULT OF PUNCH STATION'

DO include WITH mes,p_hr,p_mc,p_man,mc_hr,man_hr,man_p,to_cumT
ELSE
RETURN
ENDIF  && pub_yn

IF go_prn = .T.
EJECT
ENDIF

RELEASE ALL LIKE accum
RELEASE ALL LIKE to_cum

SET DEVICE TO SCREEN

STORE .F. TO gc_prn

WAIT

*-----BEGIN PLAN PRESS BAKE MACHINE -----*

RELEASE WINDOW pa_wind

DEFINE WINDOW pa_wind FROM 4,0 TO 22,79 ;

TITLE 'ID CODE  PD tL  tA  tK  tC  t24  cumL  cumA  cumK  cumC  CUM.T';

SYSTEM GROW ZOOM COLOR SCHEME 10

ACTIVATE WINDOW pa_wind

b_hr   = 0.0
b_mc   = 0
b_man  = 0
```

```
bb_min = 0.0
over_b = 0
CLEAR
@ 3,10 SAY 'BEGIN TO PLAN JOB IN PRESS BAKE MACHINE LINE'
@ 4,10 SAY REPLICATE('=',60)
@ 5,10 SAY 'HOW MANY HOURS DO YOU HAVE ?' GET b_hr
@ 5,60 SAY 'HOURS'
@ 6,10 SAY 'HOW MANY MACHINE DO YOU WORK ?' GET b_mc
@ 7,10 SAY 'HOW MANY MEN      DO YOU WORK ?' GET b_man
READ
bb_min = (b_hr * 60 ) * b_mc
WAIT
CLEAR
DO yesno
IF pub_yn
DO set_prn
IF go_prn = .T.
@ 0,4 SAY 'JOB PLANNING OF PRESS BAKE STATION'
@ 2,0 SAY "ID NAME                PRODUCE DWG PRG   M/C   M/C MAN  "
@ 3,0 SAY "                        (PCS) 04647           (MIN)(MIN)"
@ 4,0 SAY REPLICATE('-',76)
line = 5
ELSE
line = 0
ENDIF
SELECT partx
GO TOP
x      = 1
z      = 1
```

```
accumV = 0
accumA = 0
accumK = 0
accumC = 0
accum2 = 0
accumBM = 0  && {collect cum MAN POWER OF PRESS BAKE}
man_b. = 0  && {collect      MAN POWER}
to_cumV = 0
to_cumA = 0
to_cumK = 0
to_cumC = 0
to_cum2 = 0
b_set1 = 0
b_set2 = 0
b_set3 = 0
b_set4 = 0
b_set5 = 0
counter = 1
=ASORT(sort_part,3)
DO WHILE x <= stop_1
sum_bak = 0
man_bak = 0
IF EMPTY(sort_part(x,1)) # .T.
LOCATE FOR code = sort_part(x,1)
IF sort_part(x,2) < 0
accumV = (ABS(sort_part(x,2)) * b_lvd) + bs_lvd
accumA = (ABS(sort_part(x,2)) * b_ama) + bs_ama
accumK = (ABS(sort_part(x,2)) * b_kle) + bs_kle
accumC = (ABS(sort_part(x,2)) * b_cha) + bs_cha
```



```
accum2 = (ABS(sort_part(x,2)) * b_24 ) + bs_24
b_set1 = b_set1 + bs_lvd
b_set2 = b_set2 + bs_ama
b_set3 = b_set3 + bs_kle
b_set4 = b_set4 + bs_cha
b_set5 = b_set5 + bs_24

ELSE

accumV = 0
accumA = 0
accumK = 0
accumC = 0
accum2 = 0

ENDIF

over_b = over_b + (accumV + accumA + accumK + accumC + accum2)
accumBM = accumBM + ((accumV + accumA + accumK + accumC + accum2)* m_bak )
IF accumV # 0 .OR. accumA # 0 .OR. accumK # 0 .OR. accumC # 0
@ line,0 SAY ALLTRIM(STR(counter))
counter = counter + 1
IF go_prn = .F.
@ line,3 SAY ALLTRIM(sort_part(x,1))
a_prn = 0
ELSE
@ line,3 SAY NAME
a_prn = 30
ENDIF
IF sort_part(x,2) <= 0
@ line,12 + a_prn SAY ALLTRIM(STR(ABS(sort_part(x,2))))
ELSE
@ line,12 + a_prn SAY 0
```

```
ENDIF
IF go_prn = .T.
@ line,49 SAY ALLTRIM(SUBSTR(drawing,6,3))
ENDIF
IF go_prn = .F.
@ line,17 SAY LTRIM(STR(accumV))
@ line,21 SAY LTRIM(STR(accumA))
@ line,25 SAY LTRIM(STR(accumK))
@ line,29 SAY LTRIM(STR(accumC))
@ line,33 SAY LTRIM(STR(accum2))
ENDIF
to_cumV = to_cumV + accumV
to_cumA = to_cumA + accumA
to_cumK = to_cumK + accumK
to_cumC = to_cumC + accumC
to_cum2 = to_cum2 + accum2
IF accumV # 0
DO say_data WITH line,40,to_cumV
DO prnldata WITH line,58,'V'
pick1 = to_cumV
ELSE
@ line,40 SAY ' '
pick1 = 0
ENDIF
IF accumA # 0
DO say_data WITH line,46,to_cumA
DO prnldata WITH line,59,'A'
pick2 = to_cumA
ELSE
```

```
@ line,46 SAY ' '
pick2 = 0
ENDIF

IF accumK # 0
DO say_data WITH line,52,to_cumK
DO prnldata WITH line,60,'K'
pick3 = to_cumK
ELSE
  @ line,52 SAY ' '
  pick3 = 0
ENDIF

IF accumC # 0
DO say_data WITH line,58,to_cumC
DO prnldata WITH line,61,'C'
pick4 = to_cumC
ELSE
  @ line,58 SAY ' '
  pick4 = 0
ENDIF

IF accum2 # 0
DO say_data WITH line,64,to_cum2
DO prnldata WITH line,62,'2'
pick5 = to_cum2
ELSE
  @ line,64 SAY ' '
  pick5 = 0
ENDIF

sum_bak = (accumV + accumA + accumK + accumC + accum2)
man_bak = sum_bak * m_bak
```

```
DO CASE
CASE pick1 >= pick2 .AND. pick1 >= pick3 .AND. pick1 >= pick4
STORE pick1 TO sort_part(x,3)
DO prn_data WITH line,68,sum_bak
CASE pick2 >= pick1 .AND. pick2 >= pick3 .AND. pick2 >= pick4
STORE pick2 TO sort_part(x,3)
DO prn_data WITH line,68,sum_bak
CASE pick3 >= pick1 .AND. pick3 >= pick2 .AND. pick3 >= pick4
STORE pick3 TO sort_part(x,3)
DO prn_data WITH line,68,sum_bak
CASE pick4 >= pick1 .AND. pick4 >= pick2 .AND. pick4 >= pick3
STORE pick4 TO sort_part(x,3)
DO prn_data WITH line,68,sum_bak
CASE pick5 >= pick1 .AND. pick5 >= pick2 .AND. pick5 >= pick3
STORE pick5 TO sort_part(x,3)
DO prn_data WITH line,68,sum_bak
ENDCASE

@ LINE,73 SAY LTRIM(STR(man_bak))

line = line + 1
IF go_prn = .F.
IF line >= 14
WAIT
CLEAR
line = 0
ENDIF
ENDIF
ENDIF && {accumA..accum2 = 0}
ENDIF && {sort_part(x,1) # .T.}
x = x + 1
```



```
IF over_b >= (bb_min * z)
@ line,0 SAY REPLICATE('-',76)
line = line + 1
z = z + 1
ENDIF
ENDDO && stop_l
man_b = (accumBM / 60) / b_hr
mc_hr = 0
mc_hr = over_b / 60
man_hr = 0
man_hr = accumBM / 60
nam1 = 'LVD'
nam2 = 'AMADA'
nam3 = 'KLEEN'
nam4 = 'CHAI.'
nam5 = '24" '
mes = 'PRESS ANY KEY TO CONCLUDE RESULT OF PRESS BAKE STATION'
DO include WITH mes,b_hr,b_mc,b_man,mc_hr,man_hr,man_b,to_cumV
ELSE
RETURN
ENDIF && pub_yn
IF go_prn = .T.
EJECT
ENDIF
SET DEVICE TO SCREEN
STORE .F. TO go_prn
WAIT
*-----BEGIN PLAN ARC & SOLDER -----*
RELEASE WINDOW pa_wind
```

```

DEFINE WINDOW pa_wind FROM 4,0 TO 22,79 ;
TITLE 'ID CODE PD tARC tSOL tSPOT tFIN. cumA cumS cumW cumF';
SYSTEM GROW ZOOM COLOR SCHEME 10

ACTIVATE WINDOW pa_wind

a_hr = 0.0

a_man = 0

ab_min = 0.0

over_a = 0

CLEAR

@ 3,10 SAY 'BEGIN TO PLAN JOB IN ARC & SOLDER LINE'

@ 4,10 SAY REPLICATE('=',50)

@ 5,10 SAY 'HOW MANY HOURS DO YOU HAVE ?' GET a_hr

@ 5,60 SAY 'HOURS'

@ 6,10 SAY 'HOW MANY MEN DO YOU WORK ?' GET a_man

READ

ab_min = (a_hr * 60 ) * a_man

WAIT

CLEAR

DO yesno

IF pub_yn

DO set_prn

IF go_prn = .T.

@ 0,4 SAY 'JOB PLANNING OF ARC & SOLDER LINE '
@ 2,0 SAY "ID NAME PRODUCE DWG PROCESS TIME TOTAL"
@ 3,0 SAY " (PCS) 04647 (MIN) "
@ 4,0 SAY REPLICATE('-',76)

line = 5

ELSE

line = 0

```

```
ENDIF
SELECT partx
GO TOP
x      = 1
z      = 1
accumAR = 0
accumSO = 0
accumSW = 0
accumFI = 0
to_cumAR = 0
to_cumSO = 0
to_cumSW = 0
to_cumFI = 0
counter = 1
=ASORT(sort_part,3)
DO WHILE x <= stop_1
sum_arc = 0
IF EMPTY(sort_part(x,1)) # .T.
LOCATE FOR code = sort_part(x,1)
IF sort_part(x,2) < 0
accumAR = (ABS(sort_part(x,2)) * arc)
accumSO = (ABS(sort_part(x,2)) * solder)
accumSW = (ABS(sort_part(x,2)) * sw)
accumFI = (ABS(sort_part(x,2)) * face)
ELSE
accumAR = 0
accumSO = 0
accumSW = 0
accumFI = 0
```

```
ENDIF
over_a = over_a + (accumAR + accumSO + accumSW + accumFI)
IF accumAR # 0 .OR. accumSO # 0 .OR. accumSW # 0 .OR. accumFI # 0
@ line,0 SAY ALLTRIM(STR(counter))
counter = counter + 1
IF go_prn = .F.
@ line,3 SAY ALLTRIM(sort_part(x,1))
a_prn = 0
ELSE
@ line,3 SAY NAME
a_prn = 30
ENDIF
IF sort_part(x,2) <= 0
@ line,12 + a_prn SAY ALLTRIM(STR(ABS(sort_part(x,2))))
ELSE
@ line,12 + a_prn SAY 0
ENDIF
IF go_prn = .F.
@ line,17 SAY LTRIM(STR(accumAR))
@ line,23 SAY LTRIM(STR(accumSO))
@ line,29 SAY LTRIM(STR(accumSW))
@ line,35 SAY LTRIM(STR(accumFI))
ELSE
@ line,49 SAY ALLTRIM(SUBSTR(drawing,6,3))
ENDIF
to_cumAR = to_cumAR + accumAR
to_cumSO = to_cumSO + accumSO
to_cumSW = to_cumSW + accumSW
to_cumFI = to_cumFI + accumFI
```



```
IF accumAR # 0
DO say_data WITH line,44,to_cumAR
DO prnldata WITH line,56,'A'
pick1 = to_cumAR
ELSE
@ line,44 SAY ' '
pick1 = 0
ENDIF

IF accumSO # 0
DO say_data WITH line,51,to_cumSO
DO prnldata WITH line,58,'S'
pick2 = to_cumSO
ELSE
@ line,51 SAY ' '
pick2 = 0
ENDIF

IF accumSW # 0
DO say_data WITH line,58,to_cumSW
DO prnldata WITH line,59,'SW'
pick3 = to_cumSW
ELSE
@ line,58 SAY ' '
pick3 = 0
ENDIF

IF accumFI # 0
DO say_data WITH line,65,to_cumFI
DO prnldata WITH line,62,'F'
pick4 = to_cumFI
ELSE
```

```
@ line,65 SAY ' '
pick4 = 0
ENDIF

sum_arc = (accumAR + accumSO + accumSW + accumFI)

DO CASE

CASE pick1 >= pick2 .AND. pick1 >= pick3 .AND. pick1 >= pick4
STORE pick1 TO sort_part(x,3)

DO prn_data WITH line,67,sum_arc

CASE pick2 >= pick1 .AND. pick2 >= pick3 .AND. pick2 >= pick4
STORE pick2 TO sort_part(x,3)

DO prn_data WITH line,67,sum_arc

CASE pick3 >= pick1 .AND. pick3 >= pick2 .AND. pick3 >= pick4
STORE pick3 TO sort_part(x,3)

DO prn_data WITH line,67,sum_arc

CASE pick4 >= pick1 .AND. pick4 >= pick2 .AND. pick4 >= pick3
STORE pick4 TO sort_part(x,3)

DO prn_data WITH line,67,sum_arc

ENDCASE

@ LINE,72 SAY LTRIM(STR(sort_part(x,3)))

line = line + 1

IF go_prn = .F.

IF line >= 12

WAIT

CLEAR

line = 0

ENDIF

ENDIF

ENDIF && {accumAR and accumSO and accumSW and accumFI = 0}

ENDIF && {sort_part(x,1) = empty will out}
```

```

x      = x + 1
IF over_a >= (ab_min * z)
@ line,0 SAY REPLICATE('-',76)
line = line + 1
z = z + 1
ENDIF
ENDDO  && stop_1
man_arc = 0
nam1   = 'ARC   '
nam2   = 'SOLDER'
nam3   = 'SPOT .'
nam4   = 'FINIS.'
IF go_prn = .F.
@ $ + 1,0 SAY REPLICATE('= ',76)
WAIT 'PRESS ANY KEY TO CONCLUDE RESULT OF ARC & SOLDER STATION'
CLEAR
ENDIF
@ $ + 1,0 SAY REPLICATE('= ',76)
@ $ + 1, 0 SAY 'HOURS/DAY ='
@ $      ,13 SAY LTRIM(STR(a_hr)) + ' HRS'
@ $      ,23 SAY 'MAN /DAY ='
@ $      ,35 SAY LTRIM(STR(a_man)) + ' MAN'
gi_mn   = a_hr * a_man
man_arc = (over_a / 60 ) / a_hr
@ $ + 1,0 SAY 'MAN - HR GIVE ='
@ $      ,20 SAY gi_mn
@ $ + 1,0 SAY REPLICATE('-',76)
@ $ + 1,0 SAY 'MAN - HR USE = '
@ $      ,20 SAY over_a / 60

```

```
@ $ + 1,0 SAY 'WORK TIME ARC (HRS)='
@ $ ,35 SAY to_cumAR/60
@ $ + 1,0 SAY 'WORK TIME SOLDER (HRS)='
@ $ ,35 SAY to_cumSO/60
@ $ + 1,0 SAY 'WORK TIME SPOT WELD. (HRS)='
@ $ ,35 SAY to_cumSW/60
@ $ + 1,0 SAY 'WORK TIME SURF. FINIH. (HRS)='
@ $ ,35 SAY to_cumFI/60
@ $ + 1,0 SAY 'MAN CAPACITY SHOULD APPROX. ='
@ $ ,35 SAYman_arc
@ $ + 1,0 SAY REPLICATE('= ',76)

ELSE

RETURN

ENDIF && pub_yn

IF go_prn = .T.

EJECT

ENDIF

SET DEVICE TO SCREEN

STORE .F. TO go_prn

WAIT

RETURN

*---<SAY_DATA>---*

PROCEDURE say_data

PARAMETER say1,say2,say3

IF go_prn = .F.

@ say1,say2 SAY LTRIM(STR(say3))

ENDIF

RETURN
```



```

*---<PRN_DATA>---*

PROCEDURE prn_data

PARAMETER prn1,prn2,prn3

IF go_prn = .T.

@ prn1,prn2 SAY LTRIM(STR(prn3))

ENDIF

RETURN

*---<PRN1DATA >---*

PROCEDURE prn1data

PARAMETER p1,p2,p3

IF go_prn = .T.

@ p1,p2 SAY LTRIM(p3)

ENDIF

RETURN

*---<COLL_DATA>-----*

PROCEDURE coll_data

PARAMETER var1,var2,var3,var4

top_po = SUBSTR(var1,1,4)  && {EG. 7205,7402}

med_po = SUBSTR(var1,5,1)  && {EG. A..K}

las_po = SUBSTR(var1,6,2)  && {01..50}

top    = SUBSTR(var1,2,3)  && {205,402}

SELECT productx

GO TOP

DO WHILE .NOT. EOF()

IF SUBSTR(pd_code,1,4) = top_po .AND. SUBSTR(pd_code,6,2) = las_po

STORE SUBSTR(option,1,1) TO poption

STORE SUBSTR(option,2,1) TO poption2

ENDIF

SKIP

```

```
ENDDO
pd_qty = 0
i = 1
SELECT usex
GO TOP
DO WHILE .NOT. EOF() && usex.dbf
STORE SUBSTR(code,2,3) TO las
STORE option TO moption
STORE used TO mused
IF med_po = usex -> pd_code .AND. top = las
IF moption = 'N' .OR. moption = poption .OR. moption = poption2
pd_qty = var2 * usex -> used
STORE code .TO var3(i,1)
STORE pd_qty TO var3(i,2)
STORE weight TO var3(i,3)
var4 = i
i = i + 1
ENDIF
ENDIF
SELECT usex
SKIP
ENDDO
IF ans2 = 'WEIGHT'
cd = .F.
po = ASCAN(var3,cd)
ok1 = po - 1
rov = ASUBSCRIPT(var3,po,1)
ok2= rov - 1
DECLARE sam(ok2,3)
```

```
=ACOPY(var3,sam,1,ok1,1)
=ASORT(sam,3)
RELEASE var3
DECLARE var3(ok2,3)
=ACOPY(sam,var3,1,ok1,1)
=ASORT(var3,3)
RELEASE sam
ENDIF
RETURN
*---<TITLE PROCEDURE>---*
PROCEDURE title
PARAMETER mem1
IF ans >= 1
@ 0,0 SAY mem1
@ 0,35 SAY CDOW(DATE())
@ 0,45 SAY DATE()
@ 0,60 SAY TIME()
@ 1,0 SAY REPLICATE('-',76)
@ 2,0 SAY 'CODE PART NAME'
@ 2,42 SAY 'JOB NO:'
@ 2,50 SAY mjob(1,1) && { SHOW JOB NO. }
IF ans >= 2
@ 2,62 SAY mjob(2,1)
IF ans >= 3
@ 2,74 SAY mjob(3,1)
IF ans >= 4
@ 2,86 SAY mjob(4,1)
IF ans >= 5
@ 2,98 SAY mjob(5,1)
```

```
IF ans >= 6
@ 2,110 SAY mjob(6,1)
IF ans >= 7
@ 2,122 SAY mjob(7,1)
IF ans >= 8
@ 2,134 SAY mjob(8,1)
ENDIF && ans = 8
ENDIF&& ans = 7
ENDIF&& ans =6
ENDIF && ans = 5
ENDIF && ans = 4
ENDIF && ans =3
ENDIF && ans = 2
ENDIF && ans = 1
IF ans >= 1
@ 3,42 SAY 'DATE : '
@ 3,50 SAY mjob(1,4) && { SHOW DATE OF EACH JOB}
IF ans >= 2
@ 3,62 SAY mjob(2,4)
IF ans >= 3
@ 3,74 SAY mjob(3,4)
IF ans >= 4
@ 3,86 SAY mjob(4,4)
IF ans >= 5
@ 3,98 SAY mjob(5,4)
IF ans >= 6
@ 3,110 SAY mjob(6,4)
IF ans >= 7
@ 3,122 SAY mjob(7,4)
```



```
IF ans >= 8
@ 3,134 SAY mjob(8,4)
      ENDIF && ans = 8
ENDIF && ans = 7
ENDIF && ans =6
ENDIF && ans = 5
ENDIF && ans = 4
ENDIF && ans =3
ENDIF && ans = 2
ENDIF && ans = 1
IF ans >= 1
@ 4,42 SAY 'MODEL : '
DO type_air WITH mjob(1,2),1  && {SHOW MODEL OF EACH PO}
IF ans >= 2
DO type_air WITH mjob(2,2),2
IF ans >= 3
DO type_air WITH  mjob(3,2),3
IF ans >= 4
DO type_air WITH  mjob(4,2),4
IF ans >= 5
DO type_air WITH mjob(5,2),5
IF ans >= 6
DO type_air WITH mjob(6,2),6
IF ans >= 7
DO type_air WITH mjob(7,2),7
IF ans >= 8
DO type_air WITH  mjob(8,2),8
IF ans >= 9
DO type_air WITH mjob(9,2),9
```

```
IF ans >= 10
DO type_air WITH mjob(10,2),10
ENDIF
ENDIF && ans = 9
ENDIF && ans = 8
ENDIF&& ans = 7
ENDIF&& ans =6
ENDIF && ans = 5
ENDIF && ans = 4
ENDIF && ans =3
ENDIF && ans = 2
ENDIF && ans = 1
IF ans >= 1
@ 5,42 SAY 'COLOR : '
@ 5,50 SAY mjob(1,5) && {SHOW COLOR EACH JOB}
IF ans >= 2
@ 5,62 SAY mjob(2,5)
IF ans >= 3
@ 5,74 SAY mjob(3,5)
IF ans >= 4
@ 5,86 SAY mjob(4,5)
IF ans >= 5
@ 5,98 SAY mjob(5,5)
IF ans >= 6
@ 5,110 SAY mjob(6,5)
IF ans >= 7
@ 5,122 SAY mjob(7,5)
IF ans >= 8
@ 5,134 SAY mjob(8,5)
```

```
ENDIF && ans = 8
ENDIF && ans = 7
ENDIF && ans =6
ENDIF && ans = 5
ENDIF && ans = 4
ENDIF && ans =3
ENDIF && ans = 2
ENDIF && ans = 1
IF ans >= 1
@ 6,36 SAY 'PRODUCT CODE:'
@ 6,50 SAY mjob(1,2)  && { SHOW PRODUCT CODE EACH JOB }
IF ans >= 2
@ 6,62 SAY mjob(2,2)
IF ans >= 3
@ 6,74 SAY mjob(3,2)
IF ans >= 4
@ 6,86 SAY mjob(4,2)
IF ans >= 5
@ 6,98 SAY mjob(5,2)
IF ans >= 6
@ 6,110 SAY mjob(6,2)
IF ans >= 7
@ 6,122 SAY mjob(7,2)
IF ans >= 8
@ 6,134 SAY mjob(8,2)
ENDIF && ans = 8
ENDIF&& ans = 7
ENDIF&& ans =6
ENDIF && ans = 5
```

```
ENDIF && ans = 4
ENDIF && ans =3
ENDIF && ans = 2
ENDIF && ans = 1
IF ans >= 1
@ 7,39 SAY 'CUSTOMER : '
@ 7,50 SAY RTRIM(mjob(1,6)) && { SHOW CUSTOMER EACH JOB }
IF ans >= 2
@ 7,62 SAY RTRIM(mjob(2,6))
IF ans >= 3
@ 7,74 SAY RTRIM(mjob(3,6))
IF ans >= 4
@ 7,86 SAY RTRIM(mjob(4,6))
IF ans >= 5
@ 7,98 SAY RTRIM(mjob(5,6))
IF ans >= 6
@ 7,110 SAY RTRIM(mjob(6,6))
IF ans >= 7
@ 7,122 SAY RTRIM(mjob(7,6))
IF ans >= 8
@ 7,134 SAY RTRIM(mjob(8,6))
ENDIF && ans = 8
ENDIF&& ans = 7
ENDIF&& ans =6
ENDIF && ans = 5
ENDIF && ans = 4
ENDIF && ans =3
ENDIF && ans = 2
ENDIF && ans = 1
```



```
IF ans >= 1
@ 8,42 SAY 'VAL      : '
@ 8,50 SAY mjob(1,8)  && {SHOW VAL EACH JOB}
IF ans >= 2
@ 8,62 SAY mjob(2,8)
IF ans >= 3
@ 8,74 SAY mjob(3,8)
IF ans >= 4
@ 8,86 SAY mjob(4,8)
IF ans >= 5
@ 8,98 SAY mjob(5,8)
IF ans >= 6
@ 8,110 SAY mjob(6,8)
IF ans >= 7
@ 8,122 SAY mjob(7,8)
IF ans >= 8
@ 8,134 SAY mjob(8,8)
ENDIF && ans = 8
ENDIF&& ans = 7
ENDIF&& ans =6
ENDIF && ans = 5
ENDIF && ans = 4
ENDIF && ans =3
ENDIF  && ans = 2
ENDIF && ans = 1
IF ans >= 1
@ 9,41 SAY 'SYSTEM : '
@ 9,50 SAY RTRIM(mjob(1,9))  && { SHOW SYSTEM  EACH JOB }
IF ans >= 2
```

```
@ 9,62 SAY RTRIM(mjob(2,9))
IF ans >= 3
@ 9,74 SAY RTRIM(mjob(3,9))
IF ans >= 4
@ 9,86 SAY RTRIM(mjob(4,9))
IF ans >= 5
@ 9,98 SAY RTRIM(mjob(5,9))
IF ans >= 6
@ 9,110 SAY RTRIM(mjob(6,9))
IF ans >= 7
@ 9,122 SAY RTRIM(mjob(7,9))
IF ans >= 8
@ 9,134 SAY RTRIM(mjob(8,9))
ENDIF && ans = 8
ENDIF&& ans = 7
ENDIF&& ans =6
ENDIF && ans = 5
ENDIF && ans = 4
ENDIF && ans =3
ENDIF && ans = 2
ENDIF && ans = 1
IF ans >= 1
@ 10,40 SAY 'PACKING : '
@ 10,50 SAY mjob(1,10) && { SHOW PACKING EACH JOB }
IF ans >= 2
@ 10,62 SAY mjob(2,10)
IF ans >= 3
@ 10,74 SAY mjob(3,10)
IF ans >= 4
```

```
@ 10,86 SAY mjob(4,10)
IF ans >= 5
@ 10,98 SAY mjob(5,10)
IF ans >= 6
@ 10,110 SAY mjob(6,10)
IF ans >= 7
@ 10,122 SAY mjob(7,10)
IF ans >= 8
@ 10,134 SAY mjob(8,10)
ENDIF && ans = 8
ENDIF&& ans = 7
ENDIF&& ans =6
ENDIF && ans = 5
ENDIF && ans = 4
ENDIF && ans =3
ENDIF && ans = 2
ENDIF && ans = 1
IF ans >= 1
@ 11,40 SAY 'THERMO : '
@ 11,50 SAY mjob(1,11) && { SHOW THERMO EACH JOB }
IF ans >= 2
@ 11,62 SAY mjob(2,11)
IF ans >= 3
@ 11,74 SAY mjob(3,11)
IF ans >= 4
@ 11,86 SAY mjob(4,11)
IF ans >= 5
@ 11,98 SAY mjob(5,11)
IF ans >= 6
```

```
@ 11,110 SAY mjob(6,11)
IF ans >= 7
@ 11,122 SAY mjob(7,11)
IF ans >= 8
@ 11,134 SAY mjob(8,11)
ENDIF && ans = 8
ENDIF&& ans = 7
ENDIF&& ans =6
ENDIF && ans = 5
ENDIF && ans = 4
ENDIF && ans =3
ENDIF && ans = 2
ENDIF && ans = 1
IF ans >= 1
@ 12,37 SAY 'APPLICATION:'
@ 12,50 SAY RTRIM(mjob(1,7)) && { SHOW CUSTOMER EACH JOB }
IF ans >= 2
@ 12,62 SAY RTRIM(mjob(2,7))
IF ans >= 3
@ 12,74 SAY RTRIM(mjob(3,7))
IF ans >= 4
@ 12,86 SAY RTRIM(mjob(4,7))
IF ans >= 5
@ 12,98 SAY RTRIM(mjob(5,7))
IF ans >= 6
@ 12,110 SAY RTRIM(mjob(6,7))
IF ans >= 7
@ 12,122 SAY RTRIM(mjob(7,7))
IF ans >= 8
```



```
@ 12,134 SAY RTRIM(mjob(8,7))
ENDIF && ans = 8
ENDIF&& ans = 7
ENDIF&& ans =6
ENDIF && ans = 5
ENDIF && ans = 4
ENDIF && ans =3
ENDIF && ans = 2
ENDIF && ans = 1
IF ans >= 1
@ 13,38 SAY 'QTY (SETS):'
@ 13,53 SAY LTRIM(STR(mjob(1,3))) && { SHOW QTY EACH JOB }
IF ans >= 2
@ 13,67 SAY LTRIM(STR(mjob(2,3)))
IF ans >= 3
@ 13,78 SAY LTRIM(STR(mjob(3,3)))
IF ans >= 4
@ 13,90 SAY LTRIM(STR(mjob(4,3)))
IF ans >= 5
@ 13,101 SAY LTRIM(STR(mjob(5,3)))
IF ans >= 6
@ 13,112 SAY LTRIM(STR(mjob(6,3)))
IF ans >= 7
@ 13,123 SAY LTRIM(STR(mjob(7,3)))
IF ans >= 8
@ 13,134 SAY LTRIM(STR(mjob(8,3)))
ENDIF && ans = 8
ENDIF&& ans = 7
ENDIF&& ans =6
```

```
ENDIF && ans = 5
ENDIF && ans = 4
ENDIF && ans =3
ENDIF && ans = 2
ENDIF && ans = 1
@ 14,42 SAY'TOTAL PART'
@ 15,0 SAY REPLICATE('-',76)
RETURN
*-----< TYPE_AIR >-----*
PROCEDURE type_air
PARAMETER k1,k2
IF SUBSTR(k1,1,4) = '7205'
DO CASE
CASE SUBSTR(k1,5,1) = 'A'
@ 4,50 SAY 'ACM 10'
CASE SUBSTR(k1,5,1) = 'B'
@ 4,50 +(k2 * 12) SAY 'ACM 11'
CASE SUBSTR(k1,5,1) = 'C'
@ 4,50 +(k2 * 12) SAY 'ACM 12'
CASE SUBSTR(k1,5,1) = 'D'
@ 4,50 +(k2 * 12) SAY 'ACM 15'
CASE SUBSTR(k1,5,1) = 'E'
@ 4,50 +(k2 * 12) SAY 'ACM 16'
CASE SUBSTR(k1,5,1) = 'F'
@ 4,50 +(k2 * 12) SAY 'ACM 18'
CASE SUBSTR(k1,5,1) = 'G'
@ 4,50 +(k2 * 12) SAY 'ACM 20'
CASE SUBSTR(k1,5,1) = 'H'
@ 4,50 +(k2 * 12) SAY 'ACM 25'
```

```
CASE SUBSTR(k1,5,1) = 'J'
@ 4,50 +(k2 * 12) SAY 'ACM 30'
CASE SUBSTR(k1,5,1) = 'L'
@ 4,50 +(k2 * 12) SAY 'ACM 36'
CASE SUBSTR(k1,5,1) = 'N'
@ 4,50 +(k2 * 12) SAY 'ACM 50'
CASE SUBSTR(k1,5,1) = 'O'
@ 4,50 +(k2 * 12) SAY 'ACM 60'
ENDCASE
ENDIF
IF SUBSTR(k1,1,4) = '7402'
DO CASE
CASE SUBSTR(k1,5,1) = 'A'
@ 4,50          SAY 'MFC 200'
CASE SUBSTR(k1,5,1) = 'B'
@ 4,50 +(k2 * 12) SAY 'MFC 300'
CASE SUBSTR(k1,5,1) = 'C'
@ 4,50 +(k2 * 12) SAY 'MFC 400'
CASE SUBSTR(k1,5,1) = 'D'
@ 4,50 +(k2 * 12) SAY 'MFC 600'
CASE SUBSTR(k1,5,1) = 'F'
@ 4,50 +(k2 * 12) SAY 'MFC 900'
CASE SUBSTR(k1,5,1) = 'G'
@ 4,50 +(k2 * 12) SAY 'MFC 1000'
CASE SUBSTR(k1,5,1) = 'I'
@ 4,50 +(k2 * 12) SAY 'MFC 1200'
CASE SUBSTR(k1,5,1) = 'J'
@ 4,50 +(k2 * 12) SAY 'MFC 1600'
CASE SUBSTR(k1,5,1) = 'K'
```

```

@ 4,50 +(k2 * 12) SAY 'MFC 2000'
CASE SUBSTR(k1,5,1) = 'H'
@ 4,50 +(k2 * 12) SAY 'MFC 1000S'
ENDCASE
ENDIF
RETURN
*-----< REDUCE SUB-PROCEDURE >-----*
PROCEDURE reduce
PARAMETER v1,v2,v3,v4
IF v1 >= 1
@ line,10 SAY v2
@ line,22 SAY v3
@ line,30 SAY v4
line = line + 1
ENDIF
RETURN
*-----<INCLUDE>-----*
PROCEDURE include
PARAMETER mes,hr,mc,man,mc_hr,man_hr,man_cap,inc2,inc3,inc4,inc5,inc6,inc7
gi_mc = mc * hr
gi_mn = man * hr
IF go_prn = .F.
@ $ + 1 ,0 SAY REPLICATE('= ',76)
WAIT mes
CLEAR
ENDIF
@ $ + 1,0 SAY REPLICATE('= ',76)
@ $ + 1, 0 SAY 'HOURS/DAY ='
@ $ ,13 SAY LTRIM(STR(hr)) + ' HRS'

```



```

@ $      ,23 SAY 'M/C /DAY ='
@ $      ,35 SAY LTRIM(STR(mc)) + ' M/C'
@ $      ,45 SAY 'MAN /DAY ='
@ $      ,57 SAY LTRIM(STR(man)) + ' MAN'
@ $ + 1,0 SAY 'M/C - HR GIVE ='
@ $      ,20 SAY gi_mc
@ $      ,40 SAY 'MAN - HR GIVE ='
@ $      ,60 SAY gi_mn
@ $ + 1,0 SAY REPLICATE('-',76)
@ $ + 1,0 SAY 'M/C - HR USE = '
@ $      ,20 SAY mc_hr
@ $      ,40 SAY 'MAN - HR USE = '
@ $      ,60 SAY man_hr
@ $ + 1,0 SAY 'WORK TIME '+ nam1 + '(HRS)='
@ $      ,22 SAY inc2/60
@ $      ,39 SAY 'SETUP TIME (MIN)='
@ $      ,58 SAY LTRIM(STR(inc3))
@ $      ,63 SAY 'SETUP ='
@ $      ,71 SAY LTRIM(STR((inc3 / inc2) * 100)) + ' %'
@ $ + 1,0 SAY 'WORK TIME '+ nam2 + '(HRS)='
@ $      ,22 SAY inc4/60
@ $      ,39 SAY 'SETUP TIME (MIN)='
@ $      ,58 SAY LTRIM(STR(inc5))
@ $      ,63 SAY 'SETUP ='
@ $      ,71 SAY LTRIM(STR((inc5 / inc4) * 100)) + ' %'
@ $ + 1,0 SAY 'WORK TIME ' + nam3 + '(HRS)='
@ $      ,22 SAY inc6/60
@ $      ,39 SAY 'SETUP TIME (MIN)='
@ $      ,58 SAY LTRIM(STR(inc7))

```

```
@ $      ,63 SAY  'SETUP ='
@ $      ,71 SAY  LTRIM(STR((inc7 / inc6) * 100)) + ' %'
IF EMPTY(nam4) # .T.
@ $ + 1,0 SAY  'WORK TIME ' + nam4 + '(HRS)='
@ $      ,22 SAY  inc8/60
@ $      ,39 SAY  'SETUP TIME (MIN)='
@ $      ,58 SAY  LTRIM(STR(inc9))
@ $      ,63 SAY  'SETUP ='
@ $      ,71 SAY  LTRIM(STR((inc9 / inc8) * 100)) + ' %'
ENDIF

IF EMPTY(nam5) # .T.
@ $ + 1,0 SAY  'WORK TIME ' + nam5 + '(HRS)='
@ $      ,22 SAY  inc10/60
@ $      ,39 SAY  'SETUP TIME (MIN)='
@ $      ,58 SAY  LTRIM(STR(inc11))
@ $      ,63 SAY  'SETUP ='
@ $      ,71 SAY  LTRIM(STR((inc11 / inc10) * 100)) + ' %'
ENDIF

IF EMPTY(nam6) # .T.
@ $ + 1,0 SAY  'WORK TIME ' + nam6 + '(HRS)='
@ $      ,22 SAY  inc12/60
@ $      ,39 SAY  'SETUP TIME (MIN)='
@ $      ,58 SAY  LTRIM(STR(inc13))
@ $      ,63 SAY  'SETUP ='
@ $      ,71 SAY  LTRIM(STR((inc13 / inc12) * 100)) + ' %'
ENDIF

@ $ + 1,0 SAY  'MAN CAPACITY SHOULD APPROX. ='
@ $      ,35 SAY  man_cap * 1.1
@ $ + 1,0 SAY  'REMARK :MAN WITH 10 % ALLOWANCE FOR MATERIAL HANDLING'
```

```
@ $ + 1,0 SAY REPLICATE('= ',76)
```

```
RETURN
```

```
*---< PNEW_POX >---*
```

```
PROCEDURE pnew_pox
```

```
RELEASE WINDOW pa_wind
```

```
RELEASE WINDOW pox_wind
```

```
STORE .T. TO go_new
```

```
DO setup_pox
```

```
DO disp_pox
```

```
RETURN
```

```
*---< SET_PRN >---*
```

```
PROCEDURE set_prn
```

```
DO yesno_prn
```

```
IF pub_yn
```

```
STORE .T. TO go_prn
```

```
SET DEVICE TO PRINTER
```

```
ELSE
```

```
STORE .F. TO go_prn
```

```
SET DEVICE TO SCREEN
```

```
ENDIF
```

```
RETURN
```

```
*---< SHUT_POX >---*
```

```
PROCEDURE shut_pox
```

```
RELEASE MENU pox
```

```
RELEASE WINDOW pox_wind
```

```
RELEASE WINDOW pa_wind
```

```
SELECT partx
```

```
USE
```

```
SELECT 1
```

```
USE part INDEX part_cod ALIAS partx
```

```
SELECT partx
```

```
RETURN
```

```
*---< QUIT_POX >---*
```

```
PROCEDURE quit_pox
```

```
STORE .F. TO going
```

```
DEACTIVATE MENU pox
```

```
RETURN
```

```
*EOF*
```




ประวัติผู้เขียน

นายกิจจา ตั้งกิจวงศ์พร เกิดวันที่ 21 ตุลาคม 2508 ที่ จังหวัดกรุงเทพฯ
สำเร็จการศึกษาปริญญาวิทยาศาสตรบัณฑิต (สาขาวิศวกรรมอุตสาหการ) จากคณะ
วิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ในปีการศึกษา 2530 และ เข้าศึกษาต่อในระดับ
ปริญญาโทบัณฑิต ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ในปีการศึกษา 2533