

```
C =====  
C = ROUTINE 1.1 =  
C =====
```

```
SUBROUTINE INFLIN(LUN,LUNO,SEMAPH,LISTFI,BATOUT,CPUNO,ISTAT)
```

```
C *****  
C * ROUTINE TO GET INFORMATION AND TEXT AND WRITE THIS ON A  
C * SPECIFIED LOGICAL UNIT  
C *****
```

```
DOUBLE INTEGER BTPT  
INTEGER ENDC  
CHARACTER LPLINE * 81  
CHARACTER CPUNO * 8  
CHARACTER DELIM * 5  
CHARACTER SEMAPH * 10  
CHARACTER LISTFI * 11  
CHARACTER BATOUT * 11  
CHARACTER S3TYPE * 6  
CHARACTER ENDFI * 12  
CHARACTER SYNAM * 6
```

```
DATA DELIM          /' = '/  
DATA SYNAM          /'7ENDC='/  
DATA S3TYPE         /'**  '/
```

```
C * READ PRESENT POSITION ON THE FILE AND REWIND THE FILE
```

```
CALL REABT(LUNO,BTPT)  
REWIND LUNO
```

```
C * GET CPU-NO. CUSTOMER-NAME AND SINTRAN-VERSION
```

```
CALL GETINF(LUNO,LPLINE,ISTAT)  
IF(ISTAT.NE.0)GOTO 500
```

```
C * PUT CPU-NUMBER IN CHARACTER CPUNO
```

```
CPUNO=LPLINE(1:8)
```

```
C * GET ADDITIONAL INFORMATION
```

FABLIBR:SYMB

```
CALL ADDINF(LUNO,SEMAPH,LISTFI,BATOUT,ENDFI,LIMIT,ISTAT)
IF(ISTAT.NE.0)GOTO 500
```

```
C * GET 7ENDC
```

```
CALL ENDCOR(LIMIT,ENDC,ENDFI,CPUNO,ISTAT)
IF(ISTAT.NE.0)GOTO 500
```

```
C * WRITE THE LINE ON LOGICAL NUMBER LUN
```

```
WRITE(LUN,1000,ERR=600)LPLINE
WRITE(LUN,2000,ERR=600)SYNAM,ENDC
WRITE(LUN,1000,ERR=600)DELIM
CALL WCLOCK(LUN)
S3TYPE(3:6)=LISTFI(6:9)
WRITE(LUN,4000,ERR=600)S3TYPE
GOTO 500
```

```
C * ERROR IN WRITING ON LUN
```

```
600 IERR=ERRCODE
CALL SETBT(LUNO,BTPT)
WRITE(1,3000)IERR
CALL MON64(IERR)
```

```
500 CALL SETBT(LUNO,BTPT)
```

```
1000 FORMAT(1H$,A)
2000 FORMAT(1H$,A,Z6)
3000 FORMAT(/,1H $, *ERROR IN WRITING ON LOGG-FILE*,/,
1 1H $, *FILE SYSTEM ERROR : *,Z3,/)
4000 FORMAT(1H$,2X,A)
```

```
END
```

```
C =====
C = ROUTINE 1.2 =
C =====
```

```
SUBROUTINE APPBA(LISTFI,SEMAPH)
```

```
C *****
C * THIS ROUTINE APPENDS A BATCH-JOB TO THE BATCH-QUEUE,
C * THE BATCH-JOB IS SELECTED DEPENDING ON THE LIST-FILE-
C * NAME. THE BATCH-JOB APPENDS THE BATCH-OUTPUT-FILE
C * FOR PRESENT JOB AND CORRESPONDING LISTFILE TO THE
C * LINE-PRINTER SPOOLING QUEUE.
C *****
```

```
CHARACTER LISTFI * 11
CHARACTER JOBSEL * 9
CHARACTER CURR * 1
CHARACTER SEMAPH * 10
CHARACTER BACOM * 36
CHARACTER SLINE * 87
CHARACTER BANO * 1
CHARACTER SPACE * 1
```

```
DATA BACOM /'APP-BA (AUX)APPLIST- (SYS)DUMMY'''/
DATA JOBSEL /'123456789'/
DATA SPACE /' '/
```

```
LUN=50
```

```
C * SELECT BATCH-JOB TO BE APPENDED
```

```
CURR=LISTFI(11:11)
DO 100 I=1,9
J=I
IF(JOBSEL(I:I).EQ.CURR)GOTO 200
100 CONTINUE
GOTO 600
```

```
C * READ THE BATCH-NUMBER FROM SEMAPHORE-FILE
```

```
200 OPEN(LUN,ERR=600,FILE=SEMAPH,STATUS='OLD',ACCESS='RW')
READ(LUN,2000)SLINE
BANO=SLINE(12:12)
SLINE(12:12)=SPACE
REWIND LUN
WRITE(LUN,2000)SLINE
CLOSE (UNIT=LUN)
```

```
C * APPEND ON BATCH-PROCESS (BANO) SELECTED BATCHJOB
```

```
BACOM(8:8)=BANO
BACOM(23:23)=JOBSEL(J:J)
CALL COMND(BACOM)
GOTO 800
```

```
C *****
```

```
C * LISTFILE-NAMES NOT KNOWN TO THIS ROUTINE
```

```
600 WRITE(1,1000)
CALL COMND('OPERATOR LISTFILES CAN NOT BE APPENDED''')
```

```
C *****
C * RETURN TO CALLING PROGRAM
```

```
800 CONTINUE
```

```
C *****
```

```
1000 FORMAT(/,1H,*LISTFILES CAN NOT BE APPENDED!*,/)
2000 FORMAT(A)
```

```
END
```

```
C =====
C = ROUTINE 1.3 =
C =====
```

```
SUBROUTINE SEMOFF(SEMAPH,IDES)
```

```
C *****
C * ROUTINE TO RELEASE SEMAPHORE BY FILLING IN SPACES
C * JF IDES=1, THE ROUTINE WILL DELETE THE FILE FOUND
C * IN SEMAPHORE-FILE
C *****
```

```
IMPLICIT INTEGER (A-Z)
CHARACTER SEMAPH * 10
CHARACTER RELLIN * 120
CHARACTER DELFI * 50
CHARACTER FTYP * 6
CHARACTER FINAM * 36
CHARACTER SPACE * 1
CHARACTER YES * 1
CHARACTER NO * 1
CHARACTER ASK * 26
CHARACTER ANSW * 10
```

```
DATA DELFI /'DELETE-FILE' /
DATA FTYP /':SYMB''''/
DATA SPACE /' '/
DATA YES /'Y'/
DATA NO /'N'/
DATA ASK /'DELETING THIS FILE (Y/N)? '/
```

```
LUN=40
```

```
C * FIND FILE TO DELETE IN SEMAPHORE IF IDES=1
```

```
IF(IDES.EQ.1)THEN
OPEN(LUN,ERR=500,FILE=SEMAPH,STATUS='OLD',ACCESS='R')
READ(LUN,1000,ERR=500)RELLIN
READ(LUN,1000,ERR=500)FINAM
CLOSE(UNIT=LUN)
IF(FINAM(1:1).EQ.SPACE)GOTO 40
DO 10 I=1,30
J=I
IF(FINAM(J:J).EQ.SPACE)GOTO 20
10 CONTINUE
C * ASK IF FILE SHOULD BE DELETED
20 WRITE(1,4000)FINAM
25 WRITE(1,4100)ASK
INPUT(1)ANSW
IF(ANSW(1:1).EQ.YES)GOTO 30
IF(ANSW(1:1).EQ.NO)GOTO 40
GOTO 25
30 FINAM(J:J+5)=FTYP(1:6)
DELF(13:48)=FINAM(1:36)
CALL COMND(DELF)
```

```
ENDIF
```

```
C * PUT SPACES IN RELLIN
```

```
40 DO 50 I=1,120
RELLIN(I:I)=SPACE
50 CONTINUE
```

```
C * RELEASE SEMAPHORE FILE
```

```
OPEN(LUN,ERR=500,FILE=SEMAPH,STATUS='OLD',ACCESS='W')
WRITE(LUN,1000,ERR=500)RELLIN
CLOSE(UNIT=LUN)
GOTO 100
```

```
C *****
C * ERROR IN RELEASING SEMAPHORE
```

```
500 IERR=ERRCODE
WRITE(1,2000)SEMAPH
WRITE(1,3000)IERR
CALL MON64(IERR)
CALL COMND('OPERATOR ERROR IN RELEASING SEMAPHOREFI''')
```

```
C *****
C * RETURN TO CALLING PROGRAM
```

```
100 CONTINUE
```

```
C *****
```

```
1000 FORMAT(A)
2000 FORMAT(/,1H *ERROR IN RELEASING : *,A)
3000 FORMAT(1H ,*FILE SYSTEM ERROR : *,Z3)
4000 FORMAT(1H ,*OLD-MODEFILE NAME: *,A,/)
4100 FORMAT(1H$,A)
```

```
END
```

```
C =====
C = ROUTINE 1.4 =
C =====
```

```
SUBROUTINE ABBAT(LISTFI,SEMAPH)
```

```
C *****
C * THIS ROUTINE APPENDS A BATCH JOB TO COPY THE
C * BATCH-OUTPUT-FILE FOR THE PRESENT JOB TO THE L-P
C * THE ROUTINE ALSO EMPTIES THE CORRESPONDING
C * LISTFILE.
C *****
```

```
CHARACTER LISTFI * 11
CHARACTER JOBSSEL * 9
CHARACTER CURR * 1
CHARACTER SEMAPH * 10
CHARACTER BACOM * 33
CHARACTER BANO * 1
CHARACTER SLINE * 87
```

```
DATA JOBSSEL /'123456789'/
DATA BACOM /'APP-BA (AUX)BCOP- (SYS)DUMMY'''/
```

```
LUN=30
```

```
C * EMPTY THE LISTFILE
```

```
OPEN(LUN,ERR=500,FILE=LISTFI,STATUS='OLD',ACCESS='W')
CLOSE (UNIT=LUN)
```

```
C * SELECT BATCH-JOB TO BE APPENDED
```

```
CURR=LISTFI(11:11)
DO 100 I=1,9
  J=I
  IF(JOBSSEL(I:I) .EQ. CURR)GOTO 200
```

```
100 CONTINUE
GOTO 600
```

```
C * READ THE BATCH-NUMBER FROM THE SEMAPHORE-FILE
```

```
200 OPEN(LUN,ERR=500,FILE=SEMAPH,STATUS='OLD',ACCESS='R')
REWIND LUN
READ(LUN,4000)SLINE
CLOSE (UNIT=LUN)
BANO=SLINE(12:12)
```

```
C * APPEND ON BATCH-PROCESS (BANO) SELECTED BATCHJOB
```

```
BACOM(8:8)=BANO
BACOM(20:20)=JOBSSEL(J:J)
CALL COMND(BACOM)
GOTO 800
```

```
C *****
C * ERRORS
```

```
500 IERR=ERRCODE
WRITE(1,1000)LISTFI
WRITE(1,2000)IERR
CALL MON64(IERR)
CALL COMND('OPERATOR ERROR 1 IN ABORT''')
GOTO 800
```

```
600 WRITE(1,3000)
CALL COMND('OPERATOR ERROR 2 IN ABORT''')
```

```
C *****
C * RETURN TO CALLING PROGRAM
```

```
800 CONTINUE
```

```
C *****
```

```
1000 FORMAT(/,1H ,*ERROR IN OPENING FILE (ABORT) : *,A)
2000 FORMAT(1H ,*FILE SYSTEM ERROR : *,Z3,/)
3000 FORMAT(/,1H ,*JOB IS NOT FOUND (ABORT) *,/)
4000 FORMAT(A)
```

```
END
```

```
C =====
C = ROUTINE 1.5
C =====
```

```
SUBROUTINE CHEOCC(FINAM,CPUNO,S3TYPE,IOCCU)
```

```
C *****
C * THIS ROUTINE CHECKS IF SEMAPHORE-FILE IS USED.
C * - IF USED :
C *     WRITES THE USER, AND RETURNS IOCCU=1
C * - IF NOT USED :
C *     ASKS FOR A SYSTEM TO BE GENERATED AND
C *     RETURNS IOCCU=0
C *****
```

```
CHARACTER FINAM * 11
CHARACTER SLINE * 87
CHARACTER CPUNO * 8
CHARACTER S3TYPE * 6
CHARACTER SPACE * 1
```

```
DATA SPACE / /
```

```
LUN=10
IDEV=1
IBELL=7
IBL=37B
NORM=17B
```

```
C * OPEN THE SAAEMAPHORE-FILE AND READ THE CONTENTS
C * INTO THE CHARACTER-STRING SLINE
```

```
OPEN(LUN,ERR=500,FILE=FINAM,STATUS='OLD',ACCESS='RW')
REWIND LUN
READ(LUN,1000,ERR=500)SLINE
CLOSE(UNIT=LUN)
```

```
C * CHECK IF SEMAPHORE IS USED
```

```
IF(SLINE(1:1).NE.SPACE
1 .OR.SLINE(14:14).NE.SPACE
2 .OR.SLINE(59:59).NE.SPACE)
3 GOTO 100
```

```
C * SEMAPHORE IS NOT USED
```

```
IOCCU=0
CALL OUTCH(1,7)
WRITE(1,2000)
GOTO 800
```

```
C * SEMAPHORE IS USED
```

```
C *****
```

```
C * CHECK IF JOB IS READY FOR DUMPING
```

```
100 IF(SLINE(12:12).NE.SPACE)GOTO 300
```

```
C * WRITE WHICH JOB IS TO BE DUMPED
```

```
IF(S3TYPE.NE.SLINE(82:87))GOTO 400
IOCCU=1
CPUNO=SLINE(1:8)
CALL OUTCH(1,14B)
CALL OUTCH(1,31B)
WRITE(1,3000)
WRITE(1,3100)SLINE(1:8)
WRITE(1,3200)SLINE(14:53)
WRITE(1,3300)SLINE(59:77),SLINE(82:87)
WRITE(1,3400)
CALL WCLOCK(IDEV)
WRITE(1,3500)
GOTO 800
```

```
C *****
```

```
C * JOB IS STILL ACTIVE
```

```
300 CPUNO=SLINE(1:8)
WRITE(1,2100)IBELL,IBL,CPUNO,NORM
IOCCU=3
GOTO 800
```

```
C *****
```

```
C * NOT RIGHT TYPE OF SYSTEM
```

```
400 IOCCU=2
GOTO 800
```

```
C *****
```

```
C * ERROR IN ACCESSING SEMAPHORE
```

```
500 IFERR=ERRCODE
WRITE(1,4000)FINAM
```

WRITE(1,4100)IERR  
CALL MON64(IERR)

C \*\*\*\*\*

C \*\*\*\*\*

C === RETURN TO CALLING PROGRAM ===

C \*\*\*\*\*

800 CONTINUE

C \*\*\*\*\*

1000 FORMAT(A)

2000 FORMAT(/,1H ,\*PLEASE GENERATE THE SYSTEM BEFORE DUMPING!\*,/)

2100 FORMAT(/,1H ,A2,\*JOB FOR \*,A2,A,A2,\* IS STILL WORKING!\*,/)

3000 FORMAT(/,1H ,\*DUMP OF S-III-SYSTEM FOR :\*,/)

3100 FORMAT(1H ,\*CPU-NO : \*,A)

3200 FORMAT(1H ,\*CUSTOMER : \*,A)

3300 FORMAT(1H ,\*S-III : \*,A,2X,A,/)

3400 FORMAT(1H\$,\*DATCL : \*)

3500 FORMAT(1H ,/)

4000 FORMAT(/,1H ,\*ERROR IN ACCESSING FILE : \*,A)

4100 FORMAT(1H ,\*FILE SYSTEM ERROR : \*,Z3,/)

END

C =====

C = ROUTINE 1.1.1 =

C =====

SUBROUTINE GETINF(LUNO,LPLINE,STATE)

C \*\*\*\*\*

C \* ROUTINE TO GET CPU-NUMBER, CUSTOMER-NAME AND

C \* SINTRAN VERSION NUMBER FROM BATCH-INPUT-FILE.

C \* THE INFORMATION IS RETURNED IN THE PARAMETRE LPLINE.

C \*\*\*\*\*

INTEGER STATE,TAB1,TAB2,CPUNO,CUSTN,VERSI  
INTEGER DEL1,DEL2,DEL3

CHARACTER LPLINE \* 81

CHARACTER PARCHA \* 40

CHARACTER KAR1 \* 1

CHARACTER KAR2 \* 1

CHARACTER STARS \* 30

CHARACTER STAR \* 1

CHARACTER DOT \* 1

CHARACTER SPACE \* 1

CHARACTER LF \* 1

CHARACTER CR \* 1

CHARACTER DELIM \* 5

C \* LOCAL CONSTANTS \*\*\*\*\*

DATA STARS /'\*\*\*\*\*E\*R\*R\*O\*R\*\*\*\*\*'/

DATA STAR /'\*/

DATA DOT /'./

DATA SPACE /' '/

DATA DELIM /' = '/

CPUNO = 8

CUSTN = 40

VERSI = 19

TAB1 = 13

TAB2 = 58

DEL1 = 9

DEL2 = 54

DEL3 = 77

C \*\*\*\*\*

C \* CLEAR LPLINE \*\*\*\*\*

DO 10 I=1,81

LPLINE(I:I)=DOT

10 CONTINUE

C \*\*\*\*\*

C \* GET INFORMATION AND TEXT FROM THE TEXT-HEADER

C \*\*\*\*\*

C \* CPU-NUMBER \*\*\*\*\*

CALL FCHAR(LUNO,STAR,STATE)

IF(STATE.NE.0)GOTO 600

KAR1=STAR

CALL UNEQCH(LUNO,KAR1,STATE)

IF(STATE.NE.0)GOTO 600

KAR2=SPACE

IF(KAR1.EQ.SPACE)CALL UNEQCH(LUNO,KAR2,STATE)

IF(STATE.NE.0)GOTO 600

CALL GETCHA(LUNO,KAR2,PARCHA,CPUNO,SPACE)

```

IF (PARCHA(9:9).NE.SPACE)GOTO 600
IF (PARCHA(CPUNO:CPUNO).EQ.SPACE)GOTO 600
DO 20 I=1,8
  LPLINE(I:I)=PARCHA(I:I)
20 CONTINUE
LPLINE(DEL1:DEL1+4)=DELIM

C * CUSTOMER NAME *****
KAR1=SPACE
CALL UNEQCH(LUNO,KAR1,STATE)
IF(STATE.NE.0)GOTO 600

CALL GETCHA(LUNO,KAR1,PARCHA,CUSTN,STAR)
IF(PARCHA(CUSTN:CUSTN).NE.STAR)GOTO 600
DO 30 I=1,CUSTN-1
  INDEX=I+TAB1
  LPLINE(INDEX:INDEX)=PARCHA(I:I)
30 CONTINUE
LPLINE(DEL2:DEL2+4)=DELIM

C * VERSION NAME *****
KAR1=STAR
CALL UNEQCH(LUNO,KAR1,STATE)
IF(STATE.NE.0)GOTO 600
CALL FCHAR(LUNO,STAR,STATE)
IF(STATE.NE.0)GOTO 600
KAR1=STAR
CALL UNEQCH(LUNO,KAR1,STATE)
IF(STATE.NE.0)GOTO 600
KAR2=SPACE
IF(KAR1.EQ.SPACE)CALL UNEQCH(LUNO,KAR2,STATE)
IF(STATE.NE.0)GOTO 600

CALL GETCHA(LUNO,KAR2,PARCHA,VERSI,STAR)
IF(PARCHA(19:19).NE.SPACE)GOTO 600
DO 40 I=1,19
  INDEX=I+TAB2
  LPLINE(INDEX:INDEX)=PARCHA(I:I)
40 CONTINUE
LPLINE(DEL3:DEL3+4)=DELIM

C *****
GOTO 500

C * ERROR MESSAGES *****
600 WRITE(1,1000)STARS
WRITE(1,2000)
WRITE(1,1000)STARS

C STATE=1

1000 FORMAT(/,1H ,A,/)
2000 FORMAT(1H ,*ILLEGAL FORMAT IN HEADER OF BATCH-FILE*,/,
1 1H ,*INFORMATION CAN NOT BE FOUND OR ACCEPTED*,/,
2 1H ,* ### JOB IS ABORTED! ###*)
C *****

500 CONTINUE
END

C =====
C = ROUTINE 1.1.2 =
C =====

SUBROUTINE ADDINF(LUN,SEMAPH,LISTFI,BATOUT,ENDFI,LIMIT,ISTAT)

C *****
C * ROUTINE TO GET SEMAPHORE-FILE-NAME AND LISTFILE-NAME,
C * SYMBOL-LIST-FILE AND 7ENDC-VALUE FROM A FILE WITH
C * LOGICAL NUMBER LUN (BATCH-INPUT-FILE)
C *****

CHARACTER INFO * 40
CHARACTER SEMAPH * 10
CHARACTER LISTFI * 11
CHARACTER BATOUT * 11
CHARACTER ENDFI * 12
CHARACTER SPACE * 1
CHARACTER DELIM * 1
CHARACTER TERMC * 1
CHARACTER KAR * 1
CHARACTER STARS * 30

DATA SPACE /' '/
DATA DELIM /'%'/
DATA TERMC /'H'/
DATA STARS /'*****E*R*R*O*R*****'/

ISIZE = 40

C *****
C * FIND STARTPOINT FOR DATA

```

```
CALL FCHAR(LUN,DELIM,ISTAT)
IF(ISTAT.NE.0)GOTO 500
KAR=SPACE
CALL UNEQCH(LUN,KAR,ISTAT)
IF(ISTAT.NE.0)GOTO 500
```

```
C * GET SEMAPH , LISTFI AND BATCH-OUTPUT-FILE
```

```
CALL GETCHA(LUN,KAR,INFO,ISIZE,TERMC)
IF(INFO(12:12).NE.DELIM
1 .OR.INFO(26:26).NE.DELIM
2 .OR.INFO(40:40).NE.TERM)GOTO 500
3 GOTO 500
```

```
C * PICK SEMAPH , LISTFI AND BATCH-OUTPUT-FILE FROM INFO
```

```
SEMAPH=INFO(1:10)
LISTFI=INFO(14:24)
BATOUT=INFO(28:38)
```

```
C * GET ENDFILE **
```

```
KAR=SPACE
CALL UNEQCH(LUN,KAR,ISTAT)
IF(ISTAT.NE.0)GOTO 500
CALL GETCHA(LUN,KAR,INFO,ISIZE,TERMC)
IF(INFO(14:14).NE.TERM)GOTO 500
ENDFI=INFO(1:12)
```

```
C * GET MAX 7ENDC VALUE
```

```
CALL FCHAR(LUN,SPACE,ISTAT)
IF(ISTAT.NE.0)GOTO 500
READ(LUN,1000,ERR=500)LIMIT
```

```
GOTO 100
```

```
C * ERROR MESSAGE *****
```

```
500 WRITE(1,2000)STARS
WRITE(1,3000)
WRITE(1,2000)STARS
```

```
C *****
```

```
100 CONTINUE
```

```
1000 FORMAT(Z6)
2000 FORMAT(/,1H ,A,/)
3000 FORMAT(1H ,*ILLEGAL FORMAT! DATA CAN NOT BE FOUND!*,/,
1 1H ,* #### JOB IS ABORTED ####*)
```

```
END
```

```
C =====
C = ROUTINE 1.1.3 =
C =====
```

```
SUBROUTINE ENDCOR(LIMIT,ENDC,FINAM,CPUNO,STATUS)
```

```
C *****
C * ROUTINE TO FIND 7ENDC FROM A FILE AND CHECK
C * AGAINST LEGAL VALUES.
C *****
```

```
INTEGER ENDC,STATUS,ERROR
CHARACTER NAME * 6
CHARACTER FINAM * 20
CHARACTER SYNAM * 6
CHARACTER STARS * 30
CHARACTER OPMESS * 45
CHARACTER CPUNO * 8
```

```
DATA SYNAM /'7ENDC='/'
DATA STARS /'*****E*R*R*O*R*****'/
DATA OPMESS /'OPERATOR SYSTEM FOR IS TOO BIG!! '//'
LUN=20
STATUS=0
```

```
OPEN(LUN,ERR=500,FILE=FINAM,STATUS='OLD',ACCESS='R')
```

```
10 READ(LUN,1000,ERR=550,END=600)NAME,ENDC
IF(NAME.NE.SYNAM)GOTO 10
CLOSE(UNIT=LUN)
IF(ENDC.LT.LIMIT)GOTO 20
```

```
C * MESSAGE TO OPERATOR
```

```
OPMESS(21:28)=CPUNO
CALL COMND(OPMESS)
```

```
C * MESSAGES TO BATCH-OUTPUT FILE
```

```
WRITE(1,2000)STARS
WRITE(1,3000)SYNAM,ENDC,LIMIT
WRITE(1,2000)STARS
```



```

STATUS=1
GOTO 20

500  ERROR=ERRCODE
      WRITE(1,2000)STARS
      WRITE(1,4000)FINAM,ERROR
      CALL MON64(ERROR)
      WRITE(1,2000)STARS
      STATUS=1
      GOTO 20

550  ERROR=ERRCODE
      WRITE(1,2000)STARS
      WRITE(1,5000)FINAM,ERROR
      CALL MON64(ERROR)
      WRITE(1,2000)STARS
      STATUS=1
      GOTO 20

600  WRITE(1,2000)STARS
      WRITE(1,6000)FINAM
      WRITE(1,2000)STARS
      STATUS=1
20   CONTINUE

1000  FORMAT(A,Z6)
2000  FORMAT(/,1H ,A,/)
3000  FORMAT(1H ,*SYSTEM IS TOO BIG: *,A,Z6,/,
1      1H ,*MAX SIZE ALLOWED IS : *,Z6)
4000  FORMAT(1H ,*ERROR IN OPENING FILE: *,A,/,
1      1H ,*ERRORCODE IS : *,Z3)
5000  FORMAT(1H ,*ERROR IN READING FROM FILE: *,A,/,
1      1H ,*ERRORCODE IS : *,Z3)
6000  FORMAT(1H ,*E-O-F REACHED OF FILE : *,A,/,
1      1H ,*SYMBOL ZENDC IS NOT FOUND.*)
      FND

```

```

C =====
C = ROUTINE 1.1.4 =
C =====

```

#### SUBROUTINE WCLOCK(LUN)

```

C *****
C * ROUTINE TO WRITE DATE AND CLOCK ON
C * A LOGICAL NUMBER GIVEN FROM CALLING-
C * PROGRAM.
C *****

```

```

DIMENSION IARR(7)
CHARACTER MONTH (12) * 4

```

```

DATA MONTH(1)      /'JAN '/
DATA MONTH(2)      /'FEBR'/
DATA MONTH(3)      /'MARS'/
DATA MONTH(4)      /'APR '/
DATA MONTH(5)      /'MAI '/
DATA MONTH(6)      /'JUNI'/
DATA MONTH(7)      /'JULI'/
DATA MONTH(8)      /'AUG '/
DATA MONTH(9)      /'SEPT'/
DATA MONTH(10)     /'OKT '/
DATA MONTH(11)     /'NOV '/
DATA MONTH(12)     /'DES '/

```

```

CALL CLOCK(IARR)
I=IARR(6)
WRITE(LUN,1000)IARR(4),IARR(3),IARR(2),IARR(5),MONTH(I),IARR(7)

```

```

1000  FORMAT(1H$,J2,1H.,J2,1H.,J2,4X,I2,1X,A,2X,I4)

```

```

END

```

```

C =====
C = ROUTINE 1.1.5 =
C =====

```

#### SUBROUTINE FILINF(LUN,SEMAPH,LISTFI,BATOUT,ISTAT)

```

C *****
C * ROUTINE TO GET SEMAPHORE-FILE-NAME AND LISTFILE-NAME,
C * AND BATCH-OUTPUT-FILE FROM A FILE WITH
C * LOGICAL NUMBER LUN (BATCH-INPUT-FILE)
C *****

```

```

CHARACTER INFO * 40
CHARACTER SEMAPH * 10
CHARACTER LISTFI * 11
CHARACTER BATOUT * 11
CHARACTER SPACE * 1
CHARACTER DELIM * 1
CHARACTER TERMC * 1
CHARACTER FAR * 1

```

CHARACTER STARS \* 30

DATA SPACE            /' /'  
DATA DELIM            /'%' /'  
DATA TERMC            /' #' /'  
DATA STARS            /'\*\*\*\*\*E\*R\*R\*O\*R\*\*\*\*\*' /

ISIZE = 40

C \*\*\*\*\*

C \* FIND STARTPOINT FOR DATA

CALL FCHAR(LUN,DELIM,ISTAT)  
IF(ISTAT.NE.0)GOTO 500  
KAR=SPACE  
CALL UNEQCH(LUN,KAR,ISTAT)  
IF(ISTAT.NE.0)GOTO 500

C \* GET SEMAPH , LISTFI AND BATCH-OUTPUT-FILE

CALL GETCHA(LUN,KAR,INFO,ISIZE,TERMC)  
IF(INFO(12:12).NE.DELIM  
1            .OR.INFO(26:26).NE.DELIM  
2            .OR.INFO(40:40).NE.TERMC)  
3 GOTO 500

C \* PICK SEMAPH , LISTFI AND BATCH-OUTPUT-FILE FROM INFO

SEMAPH=INFO(1:10)  
LISTFI=INFO(14:24)  
BATOUT=INFO(28:38)

GOTO 100

C \* ERROR MESSAGE \*\*\*\*\*

500    WRITE(1,2000)STARS  
       WRITE(1,3000)  
       WRITE(1,2000)STARS

C \*\*\*\*\*

100 CONTINUE

1000    FORMAT(Z6)  
2000    FORMAT(/,1H ,A, /)  
3000    FORMAT(1H ,\*ILLEGAL FORMAT! DATA CAN NOT BE FOUND!\*, /,  
       1        1H ,\*        #### JOB IS ABORTED ####\*)

END

C =====  
C = ROUTINE 1.1.1.1                    =  
C =====

SUBROUTINE FCHAR(LUN,PCHA,STATUS)

C \*\*\*\*\*  
C \* ROUTINE TO SEARCH ON A FILE FOR A  
C \* CHARACTER GIVEN IN PARAMETER STRING.  
C \*\*\*\*\*

INTEGER STATUS  
CHARACTER PCHA \* 1  
CHARACTER KAR \* 1

EQUIVALENCE(IKAR,KAR)

STATUS=0

J=0

DO 10 I=1,5000

J=J+1

       IKAR=INCH(LUN).AND.177B  
       IKAR=ISHFT(IKAR,8)  
       IF(KAR.EQ.PCHA)GOTO 100

10        CONTINUE

          STATUS=J

100        CONTINUE

END

C =====  
C = ROUTINE 1.1.1.2                    =  
C =====

SUBROUTINE UNEQCH(LUN,PCHA,STATUS)

C \*\*\*\*\*  
C \* ROUTINE TO SEARCH ON A FILE TO FIND A  
C \* CHARACTER UNEQUAL THE CHARACTER GIVEN

```

C * IN PARAMETER STRING, THE CHARACTER
C * THAT IS FOUND IS RETURNED IN THE
C * PARAMETER STRING.
C *****

```

```

INTEGER STATUS
CHARACTER PCHA * 1
CHARACTER KAR * 1

```

```

EQUIVALENCE(IKAR,KAR)

```

```

STATUS=0

```

```

J=0
DO 10 I=1,500
J=J+1
IKAR=INCH(LUN).AND.177B
IKAR=ISHFT(IKAR,8)
IF(KAR.NE.PCHA)GOTO 100
CONTINUE
STATUS=J
100 PCHA=KAR

```

```

10
100

```

```

END

```

```

C =====
C = ROUTINE 1.1.1.3
C =====

```

```

SUBROUTINE GETCHA(LUN,PCHA,STRING,SIZE,TERMC)

```

```

C *****
C * SUBROUTINE TO GET CHARACTERS FROM A FILE AND
C * STORE THEM IN A STRING UNTIL TERMC OR MAX SIZE
C * IS REACHED. NUMBER OF CHARACTERS IS RETURNED TO
C * CALLING PROGRAM.
C *****

```

```

INTEGER SIZE,SPACE
CHARACTER STRING * 40
CHARACTER PCHA * 1
CHARACTER KAR * 1
CHARACTER TERMC * 1
CHARACTER SPACE * 1

```

```

EQUIVALENCE(IKAR,KAR)
DATA SPACE /' '/

```

```

DO 10 I=1,40
STRING(I:I)=SPACE
10 CONTINUE

STRING(1:1)=PCHA

J=1
DO 20 I=2,SIZE
J=J+1
IKAR=INCH(LUN).AND.177B
IKAR=ISHFT(IKAR,8)
STRING(I:I)=KAR
IF(KAR.EQ.TERMC)GOTO 100
20 CONTINUE

```

```

100

```

```

END
SUBROUTINE IDENT(LUN,PART)
IMPLICIT INTEGER (A-Z)

```

```

CHARACTER FILSY (6) * 70
CHARACTER RTLOA (8) * 70
CHARACTER PAONE (8) * 70

```

```

C =====
C FILESYSTEM
DATA FILSY(1) /'XXXXX XX XX XXXXX XXXXX XX XX XXXXX XX
1XXXX XXXXX XX XX'/
DATA FILSY(2) /'XX XX XX XX XX XX XX
1XX XX XXX XXX'/
DATA FILSY(3) /'XXXX XX XX XXXX XXXXX XX XX XXXXX
1XX XXXX XX X XX'/
DATA FILSY(4) /'XXXX XX XX XXXX XXXX XXXX XXXX
1XX XXXX XX XX'/
DATA FILSY(5) /'XX XX XX XX XX XX XX
1XX XX XX XX'/
DATA FILSY(6) /'XX XX XXXXX XXXXX XXXXX XX XXXXX
1XX XXXXX XX XX'/
C =====

```

```

C RT-LOADER
DATA RTLOA(1) /'XXXXX XXXXXX XX XXXX XXXX XXXX
1 XXXXX XXXX '/
DATA RTLOA(2) /'XXXXXXXX XXXXXX XX XXXXXX XXXXXX XXXXX
1 XXXXX XXXXXX'/
DATA RTLOA(3) /'XX XX XX XX XX XX XX XX

```

```

1 XX XX XX' /
  DATA RTLOA(4) /'XXXXX XX XXXX XX XX XX XX XX XX
1 XXXX XXXX '/
  DATA RTLOA(5) /'XXXXX XX XXXX XX XX XXXXXX XX XX
1 XXXX XXXX '/
  DATA RTLOA(6) /'XX XX XX XX XX XXXXXX XX XX
1 XX XX XX' /
  DATA RTLOA(7) /'XX XX XX XXXXX XXXXXX XX XX XXXXX
1 XXXX XX XX' /
  DATA RTLOA(8) /'XX XX XX XXXXX XXXX XX XX XXXX
1 XXXX XX XX' /

```

```

C =====
C PART-ONE

```

```

  DATA PAONE(1) /'XXXXX XXXX XXXXX XXXXXX XXXX XX
1XX XXXXX' /
  DATA PAONE(2) /'XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX XX
1XX XXXXX' /
  DATA PAONE(3) /'XX XX XX XX XX XX XX XX XX XXX
1XX XX '/
  DATA PAONE(4) /'XXXXXX XX XX XXXXX XX XXXX XX XX XXXX
1XX XXXX '/
  DATA PAONE(5) /'XXXXX XXXXXX XXXXX XX XXXX XX XX XXXX
1XX XXXX '/
  DATA PAONE(6) /'XX XXXXXX XX XX XX XX XX XX XX X
1XX XX '/
  DATA PAONE(7) /'XX XX XX XX XX XX XXXXXX XX
1XX XXXXX' /
  DATA PAONE(8) /'XX XX XX XX XX XXXX XX
1XX XXXXX' /

```

```

C =====

```

```

IF (PART.EQ.1) THEN
  WRITE (LUN,1000) FILSY(1)
  WRITE (LUN,1000) FILSY(1)
  WRITE (LUN,1000) FILSY(2)
  WRITE (LUN,1000) FILSY(3)
  WRITE (LUN,1000) FILSY(4)
  WRITE (LUN,1000) FILSY(5)
  WRITE (LUN,1000) FILSY(6)
  WRITE (LUN,1000) FILSY(6)
ENDIF

```

```

IF (PART.EQ.2) THEN
  WRITE (LUN,1000) RTLOA(1)
  WRITE (LUN,1000) RTLOA(2)
  WRITE (LUN,1000) RTLOA(3)
  WRITE (LUN,1000) RTLOA(4)
  WRITE (LUN,1000) RTLOA(5)
  WRITE (LUN,1000) RTLOA(6)
  WRITE (LUN,1000) RTLOA(7)
  WRITE (LUN,1000) RTLOA(8)
ENDIF

```

```

IF (PART.EQ.3) THEN
  WRITE (LUN,1000) PAONE(1)
  WRITE (LUN,1000) PAONE(2)
  WRITE (LUN,1000) PAONE(3)
  WRITE (LUN,1000) PAONE(4)
  WRITE (LUN,1000) PAONE(5)
  WRITE (LUN,1000) PAONE(6)
  WRITE (LUN,1000) PAONE(7)
  WRITE (LUN,1000) PAONE(8)
ENDIF

```

```

1000 FORMAT(1H ,A)
END
SUBROUTINE TEXT(LUN,STRING)

```

```

C *****
C ** ROUTINE TO WRITE TEXT ON PAPER-TAPE-PUNCH

```

```

  IMPLICIT INTEGER (A-Z)
  DIMENSION KAR(8,38)
  CHARACTER ALPH * 38
  CHARACTER STRING * 12

```

```

C *****

```

```

  DATA ALPH /'ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890- '/
  DATA KAR /370B,014B,012B,012B,014B,370B,000,000,
1 376B,212B,212B,212B,212B,164B,000,000,
1 070B,104B,202B,202B,202B,104B,000,000,
1 376B,202B,202B,202B,202B,174B,000,000,
1 376B,212B,212B,212B,212B,202B,000,000,
1 376B,012B,012B,012B,012B,002B,000,000,
1 070B,104B,202B,222B,222B,164B,000,000,
1 376B,010B,010B,010B,010B,376B,000,000,
1 202B,376B,202B,000,000,000,000,000,
1 140B,200B,200B,202B,202B,176B,000,000,
1 376B,010B,020B,044B,102B,200B,000,000,
1 376B,200B,200B,200B,200B,200B,000,000,
1 376B,004B,010B,010B,004B,376B,000,000,
1 376B,004B,010B,020B,100B,376B,000,000,
1

```

```

1 174B,202B,202B,202B,202B,174B,000,000,
1 376B,022B,022B,022B,022B,014B,000,000,
1 174B,202B,202B,242B,302B,374B,000,000,
1 376B,022B,022B,062B,122B,214B,000,000,
1 104B,212B,212B,212B,212B,164B,000,000,
1 002B,002B,376B,376B,002B,002B,000,000,
1 176B,200B,200B,200B,200B,176B,000,000,
1 076B,100B,200B,200B,100B,076B,000,000,
1 376B,100B,040B,040B,100B,376B,000,000,
1 302B,044B,030B,030B,044B,302B,000,000,
1 002B,004B,370B,370B,004B,002B,000,000,
1 302B,242B,222B,212B,202B,206B,000,000,
1 204B,376B,200B,000,000,000,000,000,
1 304B,242B,222B,212B,204B,300B,000,000,
1 100B,202B,212B,212B,216B,162B,000,000,
1 020B,030B,020B,024B,376B,020B,000,000,
1 100B,216B,212B,212B,212B,162B,000,000,
1 160B,210B,204B,202B,220B,140B,000,000,
1 002B,342B,022B,012B,002B,006B,000,000,
1 160B,214B,212B,212B,214B,160B,000,000,
1 014B,222B,222B,222B,122B,074B,000,000,
1 070B,104B,202B,202B,104B,070B,000,000,
1 000,010B,010B,010B,010B,010B,000,000,
1 000,000,000,000,000,000,000,000/

```

C \*\*\*\*\*

```

DO 100 I=1,12
  DO 20 J=1,38
    N=J
    IF (STRING(I:I).EQ.ALPH(N:N))GOTO 30
20  CONTINUE
30  DO 40 T=1,8
    CALL OUTCH(LUN,KAR(T,N))
40  CONTINUE
100 CONTINUE

END

SUBROUTINE TAPF(LUN,NO)
DO 10 I=1,NO
  CALL OUTCH(LUN,D)
10  CONTINUE
END
EOF

```