```
        .TITLE  IOPS INKTRONIC PRINTER HANDLER
/
/ JULY 7TH, 1971
/ LPA.
/ INKTRONIC PRINTER HANDLER(NON-API)
/ IOPS ASCII ONLY
/ KIT WHITFIELD
/
/ THIS HANDLER RECOGNISES HT(11), LF(12), VT(13), FF(14), CR(15),
/ OVERPRINT(20) AND ALT-MODE(175) AS CONTROL CHARACTERS. ALL OTHER
/ NON-PRINTING CODES ARE IGNORED.
/
/ ALT-MODE IS TREATED AS CR.
/ OVERPRINT IS TREATED AS CR IF IN FIRST POSITION IN LINE, OTHERWISE
/ IT IS IGNORED.
/
/ CR AT THE START OF A NON-NULL LINE OVERPRINTS THE LAST LINE; OTHERWISE
/ IT IS TAKEN AS A LINE TERMINATOR.  A 'LINE' CONSISTING SIMPLY OF A
/ SINGLE CR IS IGNORED.
/
/ LF AT THE START OF A LINE SPACES UP ONE LINE; ELSEWHERE IT IS IGNORED.
/
/ LF IS INSERTED AUTOMATICALLY IF THE FIRST CHARACTER OF A LINE IS
/ A PRINTING CODE.
/
/ FF IS PRINTED AS LF UNLESS AT LEAST ONE VISIBLE CHARACTER HAS BEEN
/ PRINTED SINCE THE LAST FF. INITIAL FORM FEEDS ARE TREATED LIKEWISE.
/ WHEN PRINTED AS ITSELF, FF IS TREATED AS N+6 LINE-FEEDS WHERE THERE
/ ARE 'N' AS YET UNUSED LINES ON THE PAGE.
/
/ ON CLOSING, A VT RATHER THAN A FF IS EXECUTED.
/
/
/
.MED=3
ISDF=706401                     / SKIP ON DATA FLAG
ICLP=706406                     / CLEAR FLAG, LOAD AND PRINT
ICEF=706422                     / CLEAR ERROR FLAG
ISCE=706423                     / SKIP ON AND CLEAR ERROR FLAG
ICDF=706402                     / CLEAR DATA FLAG
/
        .GLOBL  LPA.
/
        .DEC
LNLGTH=80
TABSTP=8
PGESZE=60
        .OCT
/
LPA.    DAC ARGPTR ; DAC CALPTR
        ISZ ARGPTR
        LAC* ARGPTR ; ISZ ARGPTR
        TAD (JMP SWITCH) ; DAC SWITCH
/
SWITCH  XX
        JMP INIT                / 1 = .INIT
        SKP                     / 2 = .DELET,.RENAM
```

```
        JMP IOPS06              / 3 = .SEEK
        ISZ ARGPTR              / 4 = .ENTER
        JMP* ARGPTR             / 5 = .CLEAR
        JMP CLOSE               / 6 = .CLOSE
        JMP* ARGPTR             / 7 = .MTAPE
        JMP IOPS06              / 10 = .READ
        JMP WRITE               / 11 = .WRITE
        JMP WAITR               / 12 = .WAIT,.WAITR
/
IOPS06  LAW 6 ; JMP* (.MED+1)
/
INIT    ISZ ARGPTR
        LAC (44) ; DAC* ARGPTR ; ISZ ARGPTR /     80(DEC) COLUMNS WIDE
        DAC OPEN
        DZM IOBUSY ; DZM LINCNT
CALWRD  CAL
A57     16
B57     ISDF
CHAR    LPINT
CNT5    LAC .+2
CNT7    DAC CALWRD
ACSAV   JMP* ARGPTR
/
WAITR   LAW 1000 ; AND* CALPTR
        SNA ; JMP WAIT
        LAC* ARGPTR ; AND (77777) ; DAC CALPTR
        ISZ ARGPTR
WAIT    LAC IOBUSY ; SNA ; JMP* ARGPTR
        JMP* CALPTR
/
CLOSE   LAC IOBUSY ; SZA ; JMP* CALPTR
        SAD OPEN ; JMP* ARGPTR
        LAC (CLOSE) ; DAC INTXIT
        LAW -6 ; DAC IOBUSY ; JMS CRLF
        DZM IOBUSY ; DZM OPEN
        DZM REALCH    / SO'S NEXT OUTPUT STARTS ON A BLANK 'PAGE'
        JMP CLOSE
/
WRITE   LAC IOBUSY ; SZA ; JMP* CALPTR    / WAIT TILL IT'S READY
        LAW 7000 ; DAC IOBUSY
        AND* CALPTR ; SAD (2000) ; JMP .+3
        LAW 7 ; JMP* (.MED+1)
        LAC* ARGPTR ; AND (77777) ; DAC LBH
        TAD (2) ; DAC LBP
        ISZ ARGPTR ; ISZ ARGPTR
        LAC ARGPTR ; DAC INTXIT
        LAC* LBH ; AND (377000)
        CLL!CML!CMA
        RTR ; RTR ; RTR ; RTR
        TAD (2) ; DAC TWPC
        DZM SPCNT
        LAW -1 ; DAC CHCNT ; DAC CNT5
/
GET     ISZ CNT5 ; JMP GET5
        LAC* LBP ; ISZ LBP
        ISZ TWPC ; SKP ; JMP STOP    / WORD COUNT OVERFLOW: TRUNCATE
        DAC A57
```

```
        LAC* LBP ; ISZ LBP ; ISZ TWPC
        DAC B57
        LAW -5 ; DAC CNT5
GET5    LAW -10 ; DAC CNT7
GET6    LAC B57 ; RAL
        ISZ CNT7 ; JMP GET7
        AND (177) ; JMP GOT
GET7    DAC B57
        LAC A57 ; RAL ; DAC A57
        JMP GET6
/
GOT     SZA ; SAD (177) ; JMP GET
        SAD (175) ; SKP ; SAD (20) ; LAC (15)
        DAC CHAR
        SAD (40) ; JMP SETSP
        AND (140)
        ISZ CHCNT ; SNA!CLL!CLA!CMA ; JMP FORMAT
        LAC CHCNT ; TAD (-LNLGTH-1)          / SET LINK ON LINE OVERFLOW
        LAC SPCNT ; SZL!CMA ; JMP STOP       / -> IF LINE OVERFLOW
        DAC SPCNT ; JMP .+3
        LAC (40) ; JMS NEXT          / PRECEDE NEXT REAL CHARACTER...
        ISZ SPCNT ; JMP .-3          / ...ARE 'SPCNT' SPACES
        DAC REALCH ; LAC CHAR
PRINT   JMS NEXT ; JMP GET
/
/
/
FORMAT  RAL
        LAC CHAR
        SAD (11) ; JMP SETHT
        SAD (15) ; JMP SETCR
        SZL ; JMP GET
        SAD (12) ; JMP SETLF
        SAD (13) ; JMP SETVT
        SAD (14) ; JMP SETFF
DOCRLF  LAW -1 ; JMS CRLF ; LAC CHAR ; JMP GOT / NO FORMAT: INSERT LF
/
/
/
/
SETHT   SNL ; JMP DOCRLF
        LAC CHCNT
        SZA ; TAD MINUS1 ; DAC CHCNT
        TAD (-TABSTP) ; SMA ; JMP .-2
        CMA ; TAD (1) ; DAC TEMP
        TAD SPCNT ; DAC SPCNT
        LAC TEMP ; TAD CHCNT ; DAC CHCNT
        JMP GET
/
SETSP   ISZ SPCNT ; ISZ CHCNT ; JMP GET
        ISZ CHCNT        / WE'VE JUST PUT A SPACE ON THE LINE!
/
SETLF   CLA!CMA!SKP
/
SETVT   LAW -6 ; JMS CRLF ; JMP GET
/
SETFF   LAC REALCH ; SNA!CLA!CMA ; JMP SETFF1
```

```
        LAC LINCNT
        TAD (-PGESZE+1) ; SMA!SZA ; JMP .-2
        TAD MINUS6
$ETFF1  JMS CRLF
        DZM LINCNT ; DZM REALCH
        JMP GET
/
$ETCR   LAW 774000 ; AND A57    / MASK NEXT 7-BIT CHARACTER
        SZL!SNA ; JMP STOP
        LAC (15) ; JMP PRINT    / OVERPRINT IF IN FIRST POSITION AND...
                                / ...NEXT CHARACTER IS NON-NULL.
/
/
CRLF
        DAC CNTR
CRLF1   LAC (12) ; JMS NEXT
        LAC (15) ; JMS NEXT
        ISZ LINCNT
        ISZ CNTR ; JMP CRLF1
        JMP* CRLF
CNTR
TEMP=CNTR
/
/
LPINT   DAC ACSAV
        LAC* (0) ; DAC INTXIT
        ION             /  CONSECUTIVE IOT'S ARE NON-INTERRUPTABLE
RETRY   ICDF ; ISCE ; JMP* NEXT
        LAC (.+4) ; DAC* (.MED)
        LAC (4) ; JMP* (.MED+1)
        ICEF ; JMP RETRY
/
STOP    CLA!CMA!SKP
/
NEXT    0                   / HOLDS ENTRY POINTER FOR USE ABOVE
        IOF ; DAC IOBUSY
        ISZ IOBUSY ; ICLP   / STOP AND CLEAR IOBUSY OR PRINT ANOTHER
        LAC ACSAV
        ION ; DBR ; JMP* INTXIT
/
/
/
MINUS1=DOCRLF;MINUS6=SETVT
/
INTXIT
ARGPTR
CALPTR
OPEN    0
IOBUSY  0
REALCH   0        / FIRST PAGE IS BLANK INITIALLY!
LBP
LBH
TWPC
CHCNT
SPCNT=LBH
LINCNT
        .END
```