1975

Beginners Guide to PROCSY

C. Anderson

K. Apeland

L. Breisacher

C. Canfield

T. Reiter

Report Number:
75-141

Beginners Guide to PROCSY

C. Anderson
K. Apeland
L. Breisacher
C. Canfield
T. Reiter
H. Schwetman

Computer Sciences Department

Purdue University

West Lafayette, Indiana 47907

CSD TR 141

Note: This document was prepared as part of a class project in CS 490, Spring 1975.

Beginner's Guide to PROCSY

This "Beginner's Guide to PROCSY" presents an annotated example of the use of the Purdue Remote On-Line Console System, (PROCSY) which is the terminal system provided by the Purdue University Computing Center. It is assumed that the reader is familiar with the fundamentals of using the CDC 6500-6400 system at Purdue and already has some knowledge of FORTRAN and Control Cards.

No knowledge of PROCSY or remote terminals is assumed. The reader who is trying to learn to use PROCSY is urged to work through the provided example and then to refer to other reference materials mentioned here.

In the examples that follow, the characters that the person using the terminal types are underlined. The terminal types out the rest. To correct a typing mistake, the special character, '%', acts as a backspace. For each '%' key typed, one space is erased. To erase an entire line, the 'RUBOUT' key is typed. After an entire line is correctly typed, type a 'RETURN', just as on an electric typewriter.

To start a session at the terminal, first rock the power switch at the right to ON. Check the 'ONLINE-OFF' switch is turned to 'ON LINE'. To log on to the system, press the 'CTRL' key, and while it is down, type the 'B' key. After a few seconds, the terminal will type some information as in line 1 (see example).

```
1 TCB L013 13.32.06. 03/21/75.  FULL DUPLEX
2 ACCOUNT? 77208
3 USER ID? QUH
4 PASSWORD?
5 SYSTEM?
6 PIRATE VER. 4.60
```

Then the terminal types 'ACCOUNT?' and you type your account number and hit a carriage return. The same happens with your three letter ID and password, except that when you type your password, the terminal doesn't actually type out the characters. When the terminal types out 'SYSTEM?', as in line 5, if you want PIRATE, which is used to run a typical job, simply hit a carriage return. The terminal types out line 6 and then '+++', the prompt for your command (discussed later.) The transmission mode (in the example FULL-DUPLEX) is described in PUCC document LO-PROCSY. The mode can be toggled between HALF-DUPLEX and FULL-DUPLEX by typing the CTRL-B keys during the log on.

If you need information about a command, type

+++HELP, keyword

after a command prompt, putting the command's name in as the keyword. The terminal then prints out some useful information about the command.

At the end of a session at the terminal, type

+++LOG

to log-off. The terminal will print the cost of the session. Then turn the terminal off.

```
+++LOG
TCB L130 09.52.17.  04/16/75.
ESTIMATED SESSION COST $    .29
PLEASE TURN OFF TERMINAL.  TNX.
```

CREATING A FILE

In this example a local file will be created. This example is a complete FORTRAN program with several errors which will be fixed in the EDIT segment of this document. It is possible, however, to edit as one goes along in the creation of a file. This is simply done by typing #STOP or #S. This ends the file creation. The following example contains errors which will be corrected.

```
 1  +++CREATE
 2  FILE NAME?  TEST
 3  ENTER FILE "TEST"
 4  TYPE #STOP TO TERMINATE INPUT
 5       1.000=77208,QUH.
 6       2.000=PASS-ZIP. *see below*
 7       3.000=MNF.
 8       4.000=3EOR
 9       5.000=      IMPLICIT INTEGER (A-Z)
10       6.000=      READ,A,B,C
11       7.000=      SUM=A+B;C
12       8.000=      AVE-SUM/3
13       9.000=      PRINT 10,A,B,C,SUM,AVE
14      10.000=10    FORMAT(1X,'FOR THE FOLLOIW%%WING 3 NUMBERS' ,3I4,/,1X,
15      11.000=     *       'THE SUM IS ',I4,' THE AVERAGE IS' ,I49%)
16      12.000=      STOP
17      13.000=      END
18      14.000=3%#EOR
19      15.000=10,4,22
20      16.000=#STOP
```

* Currently, this PASS card should be omitted.

lines 1-2: CREATE is a command used to create a local file. If no filename is specified as in this example, 'filename' is typed back to you as in line 2.

The filename is now TEST. The filename may have a maximum of 7 alphanumeric characters; the first must be alphabetic.

lines 3-4: messages written by the terminal.

lines 5-7: We are now ready to create our file. '1.000=' is typed out. Start typing out the control cards and then proceed on into your program. Notice that this is similar to creating a program on a key punch.

line 8: #EOR is the way to write a 7/8/9 end-of-record card. However, in this program the # was mispunched as a 3. Note this will be corrected later.

lines 11-12: Semicolon will be corrected later in EDIT to a '+', and the '-' to an '=' in 8.000.

line 14: The % key was typed twice in 'folloiw%%' to erase the last two characters. It now reads 'follo' and may be finished as shown.

line 19: data cards are typed after the #EOR.

line 20: #STOP terminates FILE CREATION.

EDITING A FILE

After looking back at the program we have written, we see there are some errors, so we use the EDIT command. (see example).

```
 1  +++EDIT
 2  #2D
 3  #4R
 4      4.000=#EOR
 5  #7R
 6      7.000=      SUM=A+B+C
 7  #8R
 8      8.000=      AVE=SUM/3
 9  #11R
10     11.000=
11     11.000=     *       'THE SUM IS' ,14, 'THE AVERAGE IS' ,14)
12  #4.51
13      4.500=C     THIS PROGRAM READS THREE NUMBERS FROM THE DATA AND
14      4.510=C     COMPUTES THE SUM AAND%%%ND ME%%VERAGE
15      4.510=C     COMPUTES THE SUM AND AVERAGE OF THEM.  THE NUMBERS
16      4.520=C     AND THEIR SUM AND ME%%AVERAGE ARE THEN PRINTED.
17      4.530=#
18  #S
    +++
```

To begin EDITing, type EDIT, "filename", where "filename" is the name of the file to be edited; if we referred to "filename" in the previous PIRATE command, then we can leave off the "filename", as on line 1, PIRATE responds with a "#", which indicates that it is ready to accept EDIT commands. (PIRATE is said to be in "EDIT mode".) First, we note that a password card is unnecessary in a program submitted from the terminal, so we DELETE line 2 of the program file by typing 2D. PIRATE takes care of the deletion, then responds with another "#". Next, we mistyped #EOR as 3EOR on line 4 of the program, so we type 4R to REPLACE line 4. PIRATE responds with the prompt for line 4, and we type in the correct line. Similarly, we mistyped lines 7 and 8, so we REPLACE those lines with the correct lines, using two 'R' EDIT commands. Trying to REPLACE line 11, we mistyped the continuation character in column 7. The RUBOUT key was typed and PIRATE gave another prompt for line 11, so we could enter it correctly. Now we realize there should be some comments in our program, so we use the INSERT command. Note that there are three places after the decimal point in the line numbers, so that it is possible to insert up to 1,999 lines between two integer numbered lines. To insert comments at the beginning of the program, we type 4.5I. PIRATE prompts us for line 4.5. After we enter line 4.5, PIRATE prompts us for line 4.51, so that several lines may be INSERTed at once. Note that line 4.51 was RUBedOUT, then entered correctly. After inserting the comments, we type "#" and PIRATE responds with the EDIT prompt, "#", again. Since we are done EDITing our program we get out of EDIT mode by typing "S" or "STOP". Some other EDIT commands are:

A - add lines with increment of 1.

C - change characters within a line.

F - find characters within a line.

Information about the use of all the EDIT commands may be found in the documents:

LO-PIRATE and LO-PIRREF.

SUBMITTING A FILE

XMIT is another Pirate command that sends a file to the computer to be executed. The file must "look" like a normal card deck including control cards, program, and data.

To use XMIT you type XMIT (see example) optionally followed by a jobname, and up to eight files. The eight files will be merged together to form one

input file. This lets you "create" and "edit" your program on one file, your data on another, and perhaps your control cards on another. After all the files are merged together the jobname is appended to the front (just like the green sequence card that you get when you read in a card deck), and the resulting file is sent off to be executed like a batch job.

XMIT does a few things to make it easier for you. First of all, you don't have to tell it which file to send (if you have only one). XMIT will use the last file name you gave to any of the Pirate commands. Also you don't have to make up a job name everytime because XMIT will make one up for you. Another thing XMIT will do is to include a password card using the password you logged. This is a good security measure and is highly recommended! Furthermore, it means that you do not have to provide a password card.

The complete form of the XMIT command is:

+++XMIT, JOBNAME,FILE1,...,FILE8

After you submit your file, it is desirable to view the output generated by your program. Pirate has another command, 'PREVIEW' which allows you to see your output before having it ROUTED or COLLECTed. It is easy to use; just type after the three pluses (+++) 'PREVIEW' (see example) followed optionally with a jobname. If no jobname is specified, then the last job sent to the computer is PREVIEWed. The COLLECT <jobname> command is also used to display output at the terminal. Once an output is COLLECTed, the ROUTE command cannot be used.

(note: for more information on PREVIEW & XMIT see LO-PIRATE or LO-PIRMAC).

```
+++XMIT
 JOB "QO" SENT
+++PREVIEW

"QO" DAYFILE:

CX       .000 SEC., NL  42100 WORDS
MNF.
CX       .001 SEC., NL  60000 WORDS
-MAX CORE NEEDED (ESTIMATE) = 051175B
LGO.
CX       .272 SEC., NL  10600 WORDS
STOP
CP       .289 SEC., IO       431 UNITS.
LINES      18
TC        2 TRACKS USED (MAX)
CM      .040 MWD-SEC.,  FL  10600 WORDS
ESTIMATED COST              .11 DOLLARS
```

"QO" OUTPUT:

```
1 06/01/74   UNIV OF MINNESOTA FORTRAN     PURDUE SINGLE PGM VERSION  MO
         MNF.
                     C       THIS PROGRAM READS THREE NUMBERS FROM THE DAT
                     C       COMPUTES THE SUM AND AVERAGE OF THEM.  THE NU
                     C       AND THEIR SUM AND AVERAGE ARE THEN PRINTED.
    000000B        1          IMPLICIT INTEGER (A-Z)
    000000B        2          READ,A,B,C
    002053B        3          SUM=A+B+C
    002055B        4          AVE=SUM/3
    002057B        5          PRINT 10,A,B,C,SUM,AVE
    002070B        6   10     FORMAT(1X,'FOR THE FOLLOWING 3 NUMBERS ',3I4,/1X
                             *       'THE SUM IS' ,I4, 'THE AVERAGE IS' ,I4)
    002070B        7          STOP
    002072B        8          END
- CTIME =      78MS  PSR LEVEL 02   CORE NEEDED (ESTIMATE) =051175B
-
0
 MNF COMPILATION COMPLETE.
-MAX CORE NEEDED (ESTIMATE) = 051175B
#EOR
 FOR THE FOLLOWING 3 NUMBERS   10   4  22
 THE SUM IS   36 THE AVERAGE IS   12
#EOR
#EOF
```

ADDITIONAL COMMANDS

GET:

```
+++GET,TEST
MAY THE EXISTING LOCAL FILE "TEST" BE DELETED? Y
 TEST          48 WORDS
```

This command gets a file out of PFILES and puts the contents in a local file of the same name. If there is already a local file of that name, PIRATE will ask if it may delete the old file before taking action. Type Y or Yes, if OK.

PUT:

```
+++PUT,TEST
 TEST          48 WORDS
```

This command takes a local file (TEST) and puts it into PFILES. Any permanent file with the same name will be automatically written over, and lost.

ROUTE:

```
+++ ROUTE Q2
  QUHQ2     = QUH0293
```

This command takes the job with the jobname "Q2" and prints it out on one of the line printers at the Math-Science Building. The job may be found in

folder 293 under "QUH0293". Output may be printed at Enad or Krannert by specifing:

+++ ROUTE Q2 AT ENAD or AT KRAN

If you want to have more security for your files, see document LO-PIRATE for information on read and write keys.

DISPLAY:

If you want to see the contents of a local file, you can use the display command to display the file. The syntax is:

```
+++DISPLAY,TEST
    1.000=77208,QUH
    3.000=MNF.
    4.000=#EOR
    4.500=C     THIS PROGRAM READS THREE NUMBERS FROM THE DATA AND
    4.510=C     COMPUTES THE SUM AND AVERAGE OF THEM.  THE NUMBERS
    4.520=C     AND THEIR SUM AND AVERAGE ARE THEN PRINTED.
    5.000=      IMPLICIT INTEGER (A-Z)
    6.000=      READ,A,B,C
    7.000=      SUM=A+B+C
    8.000=      AVE=SUM/3
    9.000=      PRINT 10,A,B,C,SUM,AVE
   10.000=10    FORMAT(1X,'FOR THE FOLLOWING 3 NUMBERS ',3I4,/,1X,
   11.000=     *       'THE SUM IS ',I4, 'THE AVERAGE IS ',I4)
   12.000=      STOP
   13.000=      END
   14.000=#EOR
   15.000=10,4,22
   16.000=#EOR
+++
```

Various options are available for controlling the number of lines printed and the format of the display. These options are detailed in the document LO-PIRATE.

Additional Comments

There are many features available to PROCSY users. In this document, we have tried to "keep it simple" and have included only enough to get you started. If there is further interest, the following documents can be referenced:

LO-PROCSY, LO-PIRATE, LO-PIRREF, LO-PED, LO-ALFIE, LO-MACROS, LO-PIRMAC, LO-PFILES.