The PC version of the SAS System is both an implementation compatible with the mainframe version, and a product reborn with significant advances in usability.

The PC version of the SAS system runs on the IBM PC XT and PC AT and compatibles running PC DOS 2.0 or higher, and having 512K memory and a hard disk.

This new system is best described by showing the screens as they appear in a real session.

When you enter the command SAS, you are greeted by our display manager with its three windows.

The most powerful tabulation procedure is TABULATE. Here we ask for a table with the rows formed by the AGE classification and ALL, and the columns formed by the mean HEIGHT and WEIGHT classified by SEX.

The CHART procedure does a variety of charts, in this case a horizontal bar chart of the means of height for each age grouped by sex.
Then we plot \textit{HEIGHT} by \textit{WEIGHT} identified by \textit{SEX}.

\begin{verbatim}
PROC GPLOT DATA=example;  
   PLOT HEIGHT*WEIGHT / GROUP=SEX;  
RUN;
\end{verbatim}

The display environment is quite powerful. Each window can be moved and sized as desired. For example, suppose that you want to "\textit{GROW}" the "\textit{TOP}" border by 5. You enter the command on the command line and hit the \texttt{ENTER} key.

\begin{verbatim}
Command:  
\end{verbatim}

Or you can hit the \texttt{cntl-G} or \texttt{cntl-S} keys to enter grow or shrink mode and use the cursor arrow keys to size the windows dynamically. Here is a side-by-side arrangement of the the windows.

\begin{verbatim}
Additional windows can be created by "pop-up" services. When you hit \texttt{cntl-K} the keyboard function definition window pops up. You can scroll through the 97 redefinable keys and interactively change their definitions in this window. You can store either display manager commands or key sequences.

To define titles, hit the \texttt{cntl-T} key. To define footnotes, hit the \texttt{cntl-F} key. Windows pop up to provide these services. Now there are six active windows. There are more windows to come...

645
If you forget what is on a data set, then you can hit \texttt{ctrl-V} to pop up the \texttt{VAR} window to see the variables on a SAS data set. This is all in living color: the \texttt{EDIT} window is green, \texttt{OUTPUT} is cyan, \texttt{LOG} is blue, \texttt{KEYS} is magenta, \texttt{FOOTNOTES} is brown, \texttt{TITLES} is green, \texttt{VAR} is black. If you want to change a window to red or any other color, you can.

Any of these windows can be zoomed by hitting the \texttt{ZOOM} function key.

The display manager commands can also be submitted from a file, or from the program editor window. In this example we ask the display manager to grow the left margin of the output window by -10, and you see the result.

If you want to run a DOS command or some other program, you can ask for it via the \texttt{X} command.

Need help to remember the features? Hit \texttt{ctrl-H}.

You can edit files, save them, then copy them back into the editor. In this case we copy the file "hilo.sas" into the editor.
This job uses the DATA step, the SAS system’s programming language for data processing. The DATA step in the PC SAS system has all the features of the mainframe version, plus a few more. With the PC version, you can store the compiled code for later execution. The PC version even compiles into Intel 8086 machine code for very fast execution. The most important new features in the DATA step are the new statements to define and use windows for data to be displayed and entered. When the HILO program is executed, it displays a window called RULES describing the game it implements.

When you guess a number too high, it displays and blinks the HIGH window requesting a lower number. When you guess too low, it tells you with the LOW window. Other windows pop up when you want to cheat, or when you guess the correct window.
Programming these windows is easy. Here is some of the code from the *HILO* game:

```
1I~:~I!~f~ co!ol'=blac~
1'0115=7 col!ulhs=28 il'OlI=l icolWlfl=1
~fr.~~),
clIllIl'=blUl! ilttl'=Liink
96x686
++ last color-::l'I!d protect:yu
" 1.12 'IS TOO HIGl!I' colol'=biUl! aUr=Llink 16 P2 '',xt Guess1' coiol'=!fl'I!en coIor-::yellow
96x669
lIindoli 1!11 color-::L1ack roIlS=' colWlfls=311 il'OlI=17 icoIWIII=5& pane:l!..,ty' ,
85x638
[Image 0x73 to 594x842]
```

Another *DATA* step program we wrote for the SUGI meeting was "sugilook", which prompted users for their last names, then did a rapid search of the 2000+ registered users to find the one specified.

```
The ability to write *DATA* step programs that are full-screen and interactive will open the way to applications where you need to acquire data in a very flexible way under program control.

Interactivity is also a desirable attribute in statistical analysis. In the PC SAS system, many statistical procedures are newly interactive. For example in GLM, we can fit some model, look at the results, then ask for more details interactively — in this case the *MEANS* command for *AGE* or a certain contrast.
Suppose you want to first read in the data to create a SAS data set called CLASS (in a manner like the DATA step), then you want to use this data set, first to list out the data (like PROC PRINT or PROC FPSTRN), then obtain some summary statistics (like PROC MEANS or PROC SUMMARY). You can do it all in the SAS/IML system as shown:

```
create class var(na! sex age height Right);
infile ·clus.dat;
do data; input	na$ sex$ age1 height wight;
end;

close file ·class.dat;
```

Suppose you want to first read in the data to create a SAS data set called CLASS (in a manner like the DATA step), then you want to use this data set, first to list out the data (like PROC PRINT or PROC FPSTRN), then obtain some summary statistics (like PROC MEANS or PROC SUMMARY). You can do it all in the SAS/IML system as shown:

```
create class var(na! sex age height Right);
infile ·clus.dat;
do data; input	na$ sex$ age1 height wight;
end;

close file ·class.dat;
```

Interactivity in procedures is limited to executing set commands; you cannot do any complex programming. The DATA step provides a complete programming environment, but it is not interactive in a programming sense. Wouldn't it be nice if you could have both a completely interactive programming environment (like BASIC), together with a powerful data processing language (like the DATA step), and even provide higher level applications like summary statistics (like Procedures)? That is the aim of a new product called the SAS/IML™ system.
And you can also use the IML system as a query facility. Suppose that you want to list all the records where NAME starts with a "J". It will respond with surprising speed if you have indexed the data set by the variable you search with. Then you want to find all the record numbers where AGE is greater than 13 into a list called P. Then you read several of the variables for these records into memory, compute a new variable RATIO from HEIGHT and WEIGHT, and print out the result.

Note that the variables you read in from multiple records become matrices, and you can perform arithmetic on these entire matrices of values. The SAS/IML system, as successor to the MATRIX procedure, can do all kinds of matrix computations. This feature is not only valuable for statistical tasks, but also for data processing tasks where you want to manipulate a set of values in a systematic way.

This is all interactive. The commands execute as soon as you enter them. And it is all programmable—every statement is executable and can be done immediately or stored in a module. Any statement can be made part of an iterative or conditional clause. SAS/IML software is becoming a total applications and programming environment.

All done with your IML session? If you want to save the current values, you just give the command STORE, and you can retrieve them in a later session.

All done with your SAS session? If you want to save a copy of the output for printing, just say “SAVE filename” on the command line of the output screen. Then say “QUIT” on any command line, and you are returned to PC DOS.

For many years the SAS System has been among the most respected applications software systems on the mainframe. Now it is available on the desktop. And it is better than ever. A new menuing facility called SAS/AF will be available on the PC version; this will help even novice and casual users to be immediately productive users of SAS.