USER'S GUIDE
Graphical User Interface for BACI

Bill Bynum/Tracy Camp
College of William and Mary/Colorado School of Mines

November 5, 2002

Contents

List of Figures

1 Introduction

The original version of the Graphical User Interface (GUI) to the Concurrent PCODE Interpreter of the BACI System was created in 1996 by Vince Gibson in partial fulfillment of the requirements for a Masters Degree in Computer Science at the University of Alabama under the direction of Tracy Camp. His design was modified and improved significantly in the summer of 1998 and 1999.

The BACI GUI program gives the user the capability to monitor all aspects of the execution of a program written in either C−− or Pascal, the languages supported by the two compilers of the BACI system. All source files used to produce the program are displayed, if they are available. The user can set breakpoints and single-step at either the source code level or the PCODE level. The user can view the program data and the PCODE instructions executed. All non-source windows can be saved to disk.

The BACI GUI program requires a UNIX operating system (Linux, Sun4, IRIX, or AIX), equipped with the X11 windowing system and the Tcl8.0 and Tk8.0 program libraries created by John Ousterhout.

Programs of the BACI System

program	function	described in
`bacc`	BACI C−− to PCODE Compiler	cmimi.ps
`bapas`	BACI Pascal to PCODE Compiler	guidepas.ps
`bainterp`	command-line PCODE Interpreter	cmimi.ps, guidepas.ps disasm.ps
`bagui`	Graphical user interface to the PCODE Interpreter (UNIX systems only)	this document (guiguide.ps)
`badis`	PCODE de-compiler	disasm.ps
`baar`	PCODE archiver	sepcomp.ps
`bald`	PCODE linker	sepcomp.ps

2 The Top Level Menu Bar

The BACI GUI program is composed of two parts: the binary executable file and a Tcl/Tk script, both of which should be located in the same directory.

When you invoke the BACI GUI from an Xterm window, you should see the top level menu bar shown in Figure 1. We discuss the File, Options, Data, and Breakpoint choices in Sections 2.1-2.4, respectively.

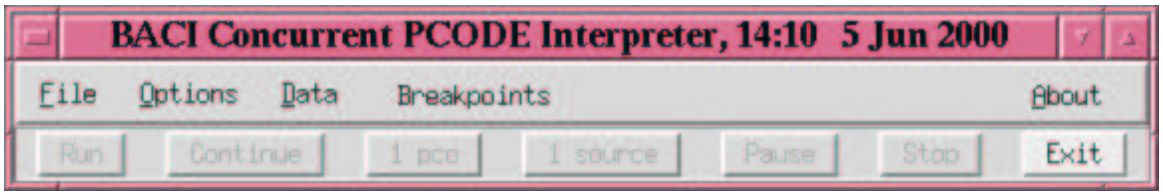

Figure 1: Top Level Menu Bar

The buttons across the bottom of the menu bar are associated with the execution of a program. At startup, all of the buttons across the bottom of the bar except the `Exit` button are disabled. To enable these buttons, you must first open a PCODE file using the `File` menu (see Section 2.1).

After a `.pco` file has been opened, the `Run` button will be the only inactive button. The `Continue` button will continue to execute the file that has been loaded until the next breakpoint is reached (see Section 4 on breakpoints), or the last PCODE instruction of the program has been executed. The `1 pco` and `1 source` buttons execute one PCODE instruction or one source line, respectively. The `Pause` button pauses program

execution. The `Stop` button terminates program execution. The `Run` button and the `Open PCODE file` item of the `File` menu (see Section 2.1 below) are both re-activated, so that either the program can be re-executed from the beginning or a new program can be loaded. The `Exit` button exits the BACI GUI program.

2.1 The File Menu

Figure 2 shows the `File` menu of the top level menu bar.

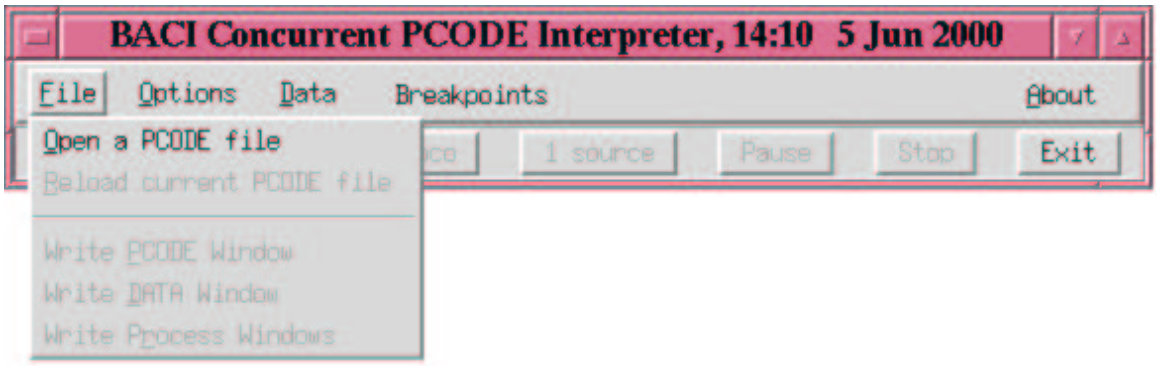

Figure 2: File Menu

As you can see in Figure 2, the only active item in the menu is `Open a PCODE file`. The `Reload current PCODE file` item is activated after a PCODE file has been loaded. This item would typically be used in the case when the program source is modified and recompiled in a separate `xterm` window. The inactive items on the menu below the separator bar are activated after their corresponding windows have been created. The `PCODE` window is created when you choose the `Debug PCODE` item from the `Options` menu (see Figure 3). Section 5 discusses the `PCODE` window. The `Data` window is created when either you choose to view program data by selecting items from either the `Data` (see Figure 4) or `Options` menus or data is generated by a runtime exception. Section 6 discusses the `Data` window. The `Process` window is created when a `PCODE` file is opened. Output generated by the process appears in this window. Section 7 discusses the `Process` window.

2.2 The Options Menu

Figure 3 shows the `Options` menu of the top level menu bar.

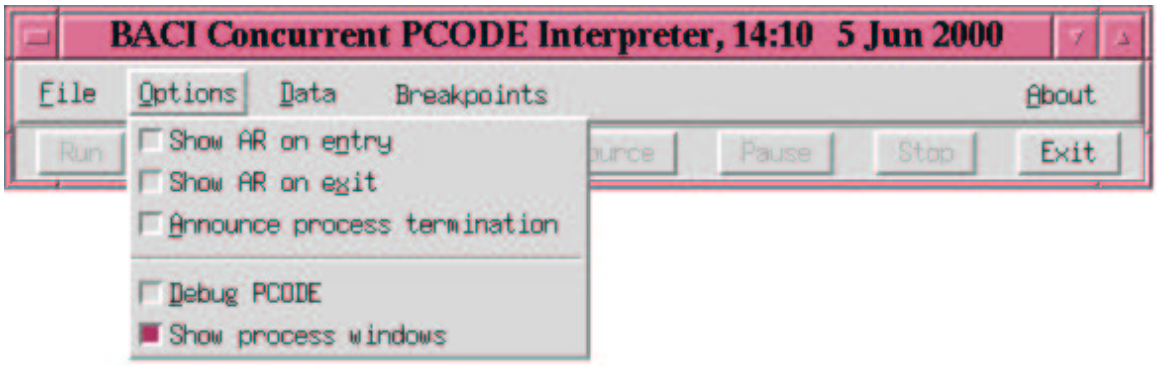

Figure 3: Options Menu

You can use this menu to select several different execution options. You can choose to display the activation record (AR) of each procedure or function on entry or exit. The `Data` window will be created to display the information. You can have the termination of each concurrent process written to the `Data` window also.

Turning on the `Debug PCODE` choice button creates the `PCODE` window and displays each subsequently executed PCODE instruction. Turning off the `Debug PCODE` choice button deletes the `PCODE` window.

By default, each concurrent BACI process is provided a separate window into which it writes its output. If the number of concurrent processes is large, then the clutter in the root window from the process windows can be undesirable. Turning off the `Show process windows` removes all process windows except the window for the main process.

2.3 The Data Menu

The `Data` menu is shown in Figure 4.

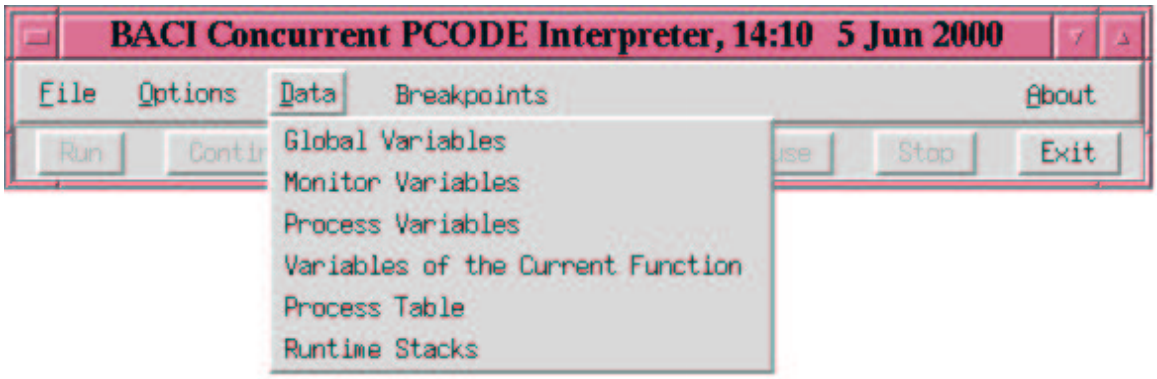

Figure 4: Data Menu

As you can see from Figure 4, you can view the current values of the global variables, the variables global to a monitor, the variables of the current processes, or the variables of the currently executing function or procedure. You can also choose to view the current state of the process table and the runtime stacks. All of this information will be displayed in the `Data` window.

Figure 13 shows an example of typical Data window usage.

2.4 The Breakpoints Button

The `Breakpoints` button in the top level menu bar (see Figure 1) is not a menu button. Clicking this button creates a window with which you can set and unset breakpoints at both the PCODE and source line level. The Breakpoints window is discussed in Section 4.

3 Selecting a PCODE File to Execute

To open a PCODE file to execute, simply select the `Open PCODE File` item of the `File` menu on the top level menu bar (see Figure 2).

Opening a PCODE file will create a file selection window like the one shown in Figure 5. The name of the current directory is shown in an editable field at the top of the window. By modifying the name of the directory shown and pressing `RETURN`, a user can move to any node in the user's directory tree.

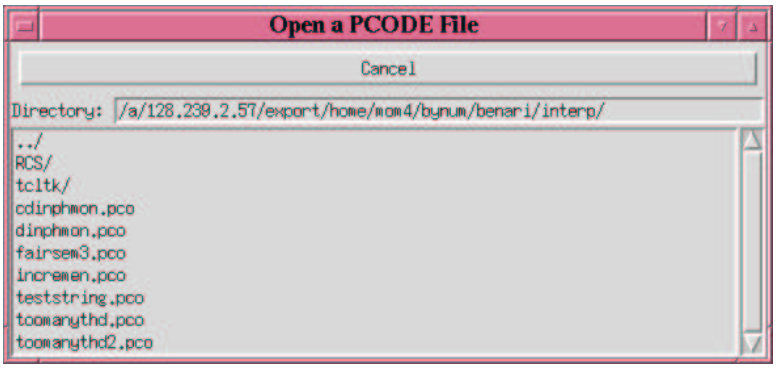

Figure 5: Open File Dialog Window

Notice that all directories are marked with trailing slash (/). Double-clicking on a directory name changes to that directory. The directory name `../` denotes the parent directory of the current directory. This provides the user an alternative means for traversing the user's directory tree.

Only files having the `.pco` suffix (the PCODE files) and directories are listed in the file selection window. When you select a PCODE file, then source windows are created to display the source files used to create the PCODE file, if the source files are available. In addition, the output window for the main program (the `Main` process window) is created.

Figure 6 shows what will happen when the `fairsem3.pco` PCODE file is selected. This PCODE file was created with three different source files, `fairsem3.pm`, `fairsem3.inc`, and `comp-swap.inc`. Each source file is displayed in a separate source window.

4 Setting and Unsetting Breakpoints

Breakpoints are manipulated through the Breakpoints window created when the `Breakpoints` button on the top of the top level menu bar (see Figure 1) is clicked. The Breakpoints window for the `fairsem3.pco` program is shown in Figure 7.

Notice that a breakpoint has been set at PCODE location 437 on line 15 of the source file `fairsem3.pm`. This breakpoint was set by one of two ways: by entering the number "437" in the entry box next to `Enter PCODE address for a breakpoint` or by entering the number "15" in the entry box next to `Enter source line number for a breakpoint`. The breakpoint is marked on the line of the appropriate source window with a `b` character on the left of the line (see Figure 11).

If you try to place a breakpoint on a line of a source file that does not generate any PCODE, then the breakpoint will actually be inserted just before the next source line that generates PCODE. In the following C−− source code fragment, if you try to insert

```
12        i = 0;
13        do {
14           /*
15              iterate over the
16              elements of the array
17           */
18           a[i] = i;
19        } while (i < max_a);
```

a breakpoint at line 15, the breakpoint will be set on line 18, because the comment spanning lines 14 through 17 does not generate any PCODE. These lines of the program source do not correspond to any executable PCODE instruction.

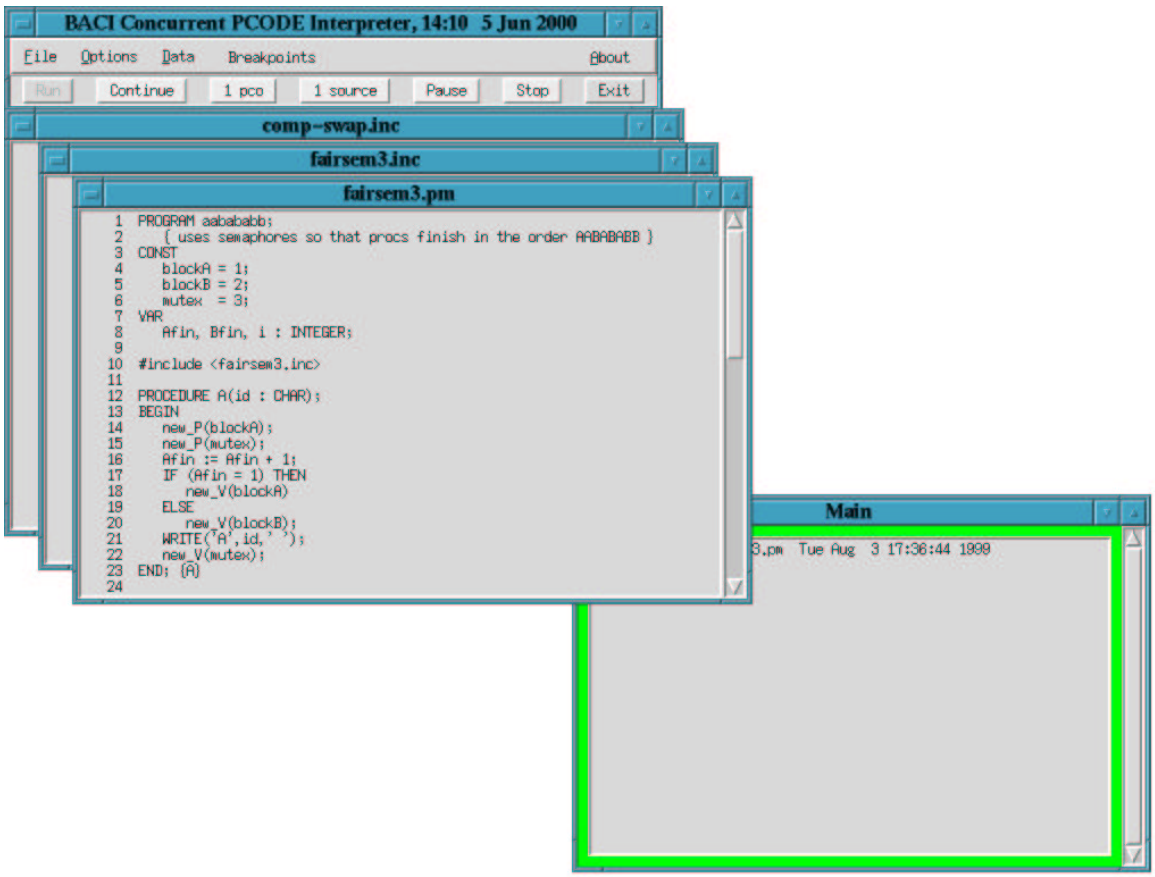

Figure 6: Root Window after Opening a `.pco` File

When more than one source file is used to create a PCODE file, as was the case for the `fairsem3.pco` file, there will be a menu button to the right of the `Source File:` label in the second pane of the Breakpoints window. The name of the currently selected file will be the title of this menu button. Clicking the menu button will pop up a list of files (see Figure 8). Initially, the primary source file is selected and its name appears on the menu button.

In Figure 8, the source file `fairsem3.inc` has just been selected. When the menu is released, the the corresponding source window will rise to the top of the window stack and the name of the chosen source file will become the label on the menu button.

When a PCODE file is produced by a single source file, the menu button to the right of the `Source File:` label is replaced by the name of the only source file.

To delete a breakpoint, simply select the breakpoint you want to delete in the listbox below the second pane of the Breakpoints window and click the `Delete selected breakpoint` button along the bottom of the window. If you click the `Delete selected breakpoint` button when there is no breakpoint selected, then the message window shown in Figure 9 will appear. After five seconds, the message window will disappear. These BACI Note windows are used in several places when there is information the user needs to have, but no user response is required.

When a program execution reaches a breakpoint that has been set, a breakpoint alert will appear. The breakpoint alert corresponding to the breakpoint on line 15 of the `fairsem3.pm` file is shown in Figure 10. This alert will remain until you click one of the other buttons on the bottom of the top level menu bar, such as `Continue` `1 pco`, `1 source`, `Stop`, or `Exit`.

When a breakpoint is reached, the source window that contains the line where the breakpoint has oc-

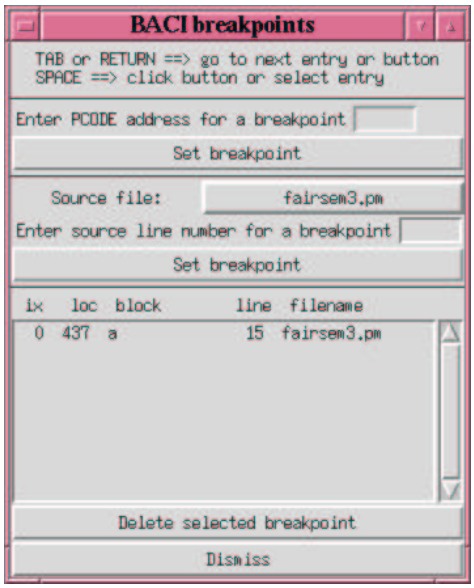

Figure 7: Breakpoints Window

curred will rise to the top of the source window stack and the line corresponding to the breakpoint will be highlighted. Figure 11 shows the source window for the `fairsem3.pm` file just after the breakpoint on line 15 has occurred.

5 The PCODE Window

The PCODE window is created when you select the `Debug PCODE` option from the `Options` menu (see Figure 3) of the top level menu bar. Figure 12 shows the PCODE window for the `fairsem3.pco` program just after the breakpoint at location 437 (line 15 of `fairsem3.pm`) has occurred.

The PCODE window is especially useful when you are using the `1 pco` button of the top level menu bar to execute one PCODE instruction at a time.

When the PCODE window exists, its contents can be written to a file with the `Write PCODE Window` entry of the `File` menu (see Figure 2) in the top level menu bar.

6 The Data Window

The Data window is created either when you select one of the items from the `Data` menu (see Figure 4) of the top level menu bar or when debugging data is generated by a run time exception.

Figure 13 shows the Data window after a dump of the global and main proc variables of the `fairsem3.pco` program. Note that the Data window is time stamped when it is opened. Each collection of data written to the Data window is identified with the number of the process running at the time of the dump, the PCODE location, the source line number, and the time and date.

7 The Process Windows

The window for the Main process is created when the PCODE file is opened. Any output generated by the main program appears in this window. Output from all of the process windows generated during program

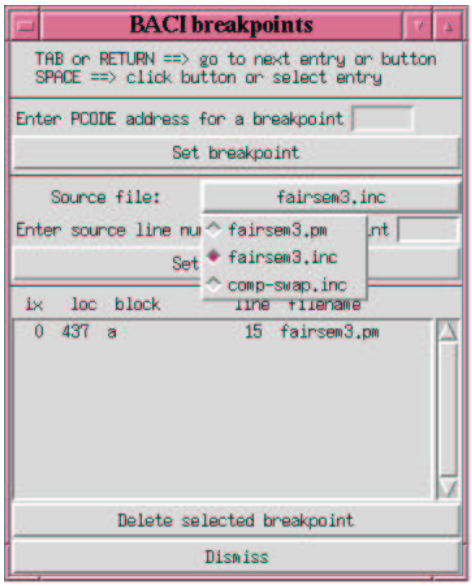

Figure 8: Source File Menu of the Breakpoint Window

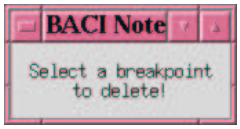

Figure 9: BACI Note Alert

execution is shown in the Main process window. Figure 14 shows the main process just after the breakpoint on line 15 of the source file `fairsem3.pm` has been reached.

The `fairsem3.pm` program creates eight concurrent processes. Each of them has its own output window. Figure 15 shows the window for process 2 when the breakpoint on line 15 of the source file `fairsem3.pm` has been reached.

If a loaded program is re-run or if a new PCODE file is loaded, then all process windows are deleted and re-created when needed as the program executes.

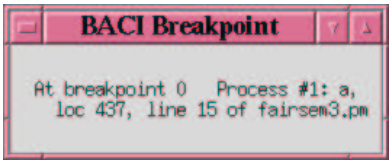

Figure 10: Breakpoint Alert

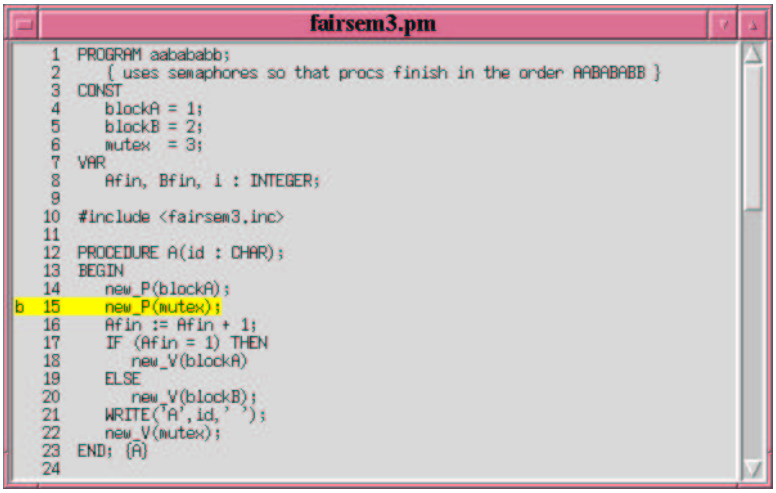

Figure 11: Source File Window at a Breakpoint

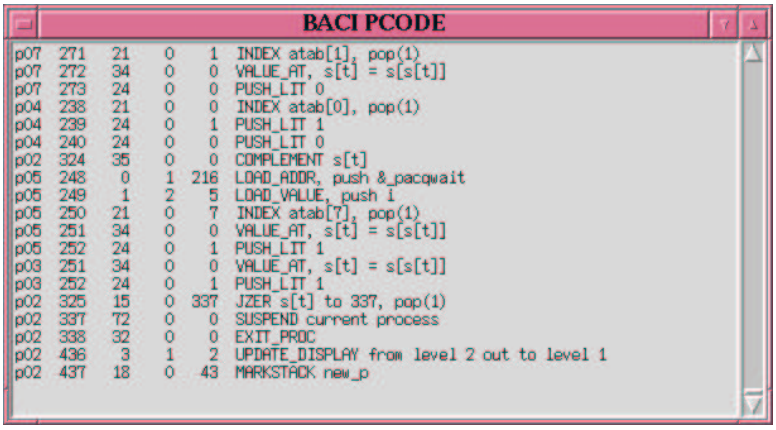

Figure 12: PCODE Window

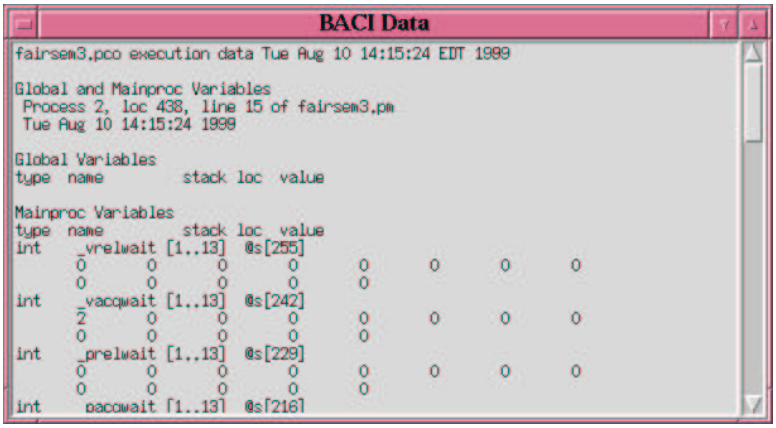

Figure 13: Data Window

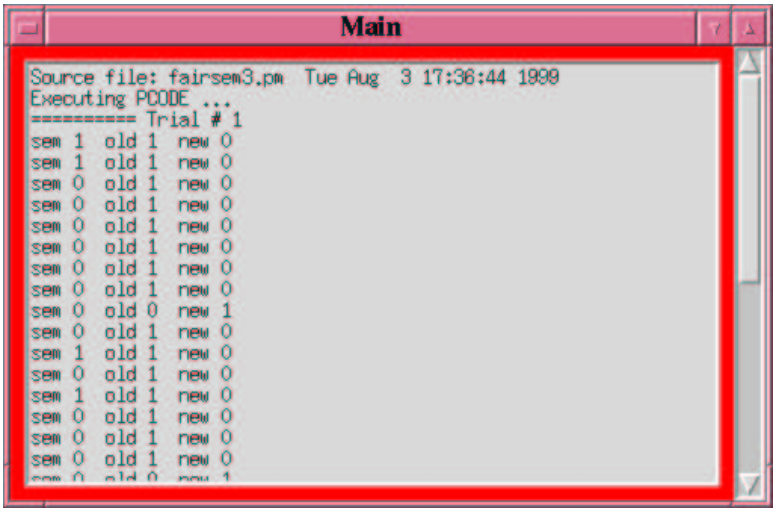

Figure 14: Main Process Window

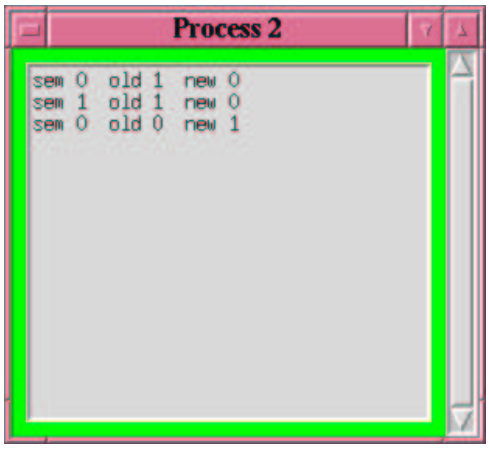

Figure 15: A Process Window