SPY

Larry Masinter
17 October 1984

SPY is a tool to help you make programs run faster. It gives you a picture of where the program is spending its time. SPY has two parts: a 'sampler' which you run while running your program, which monitors what the program is doing, and a 'displayer' which displays the data gathered by the sampler.

The 'sampler' periodically interrupts the running program to account the functions in the current call stack. This allows SPY to remember not only (proportionally) how long is spent in each individual function, but also how long each function is seen on the call stack. The sampler data structures were designed to minimize interference with the normal running of the program -- there is little noticable performance degradation. Because SPY doesn't log every call and return you can run it even over long computations, unlike PROSTATS.

The 'displayer' uses the GRAPHER package to display the data gathered by the sampler. In the graph, the height of each node is adjusted to be proportional to the amount of time. Just as Masterscope and Browser give an interactive picture of the static structure of the program, SPY gives an interactive picture of the dynamic structure. The displayer is interactive as well as graphical. That is, you can look at the data in a variety of ways, since it seems there is no one picture that says it all. Since the displayer knows the whole call graph, it can show the entire tree structure (with separate calls to a function accounted separately) or merging separate calls to the same function. Since the sampler records the entire calling stack when it samples, it can account for both individual and cumulative time. When the sampler runs, if a function is on the top of the stack, it adds to its individual total; if the function is on the stack at all, the sampler adds to the cumulative total.

When there are several calls to the same function within the graph, the displayer can either 'merge' the nodes (show the total time for the function in one node) or not. If a node is merged, then one of the boxes in the graph will have all of the time for that function accounted to it, and the rest will be left as 'ghost' boxes. SPY has a variety of ways of controlling which nodes will be merged.

How to Use SPY

(SPY.BUTTON) [Function]

This function puts up a little menu with the single button "SPY" on it. Pushing it once will turn on sampling (SPY.START). Pushing it again will turn off sampling (SPY.END) and display the results (SPY.TREE 10). This is the simplest way of spying on operations. Note that you can't turn off sampling if the mouse process is locked out. Alternatively, you can turn SPYing on and off by using the following functions:

(SPY.START) [Function]
Reinitializes the internal SPY data structures, and turns on sampling.

(SPY.END) [Function]
Turns off sampling, and cleans up the data structures.

(WITH.SPY form) [Macro]

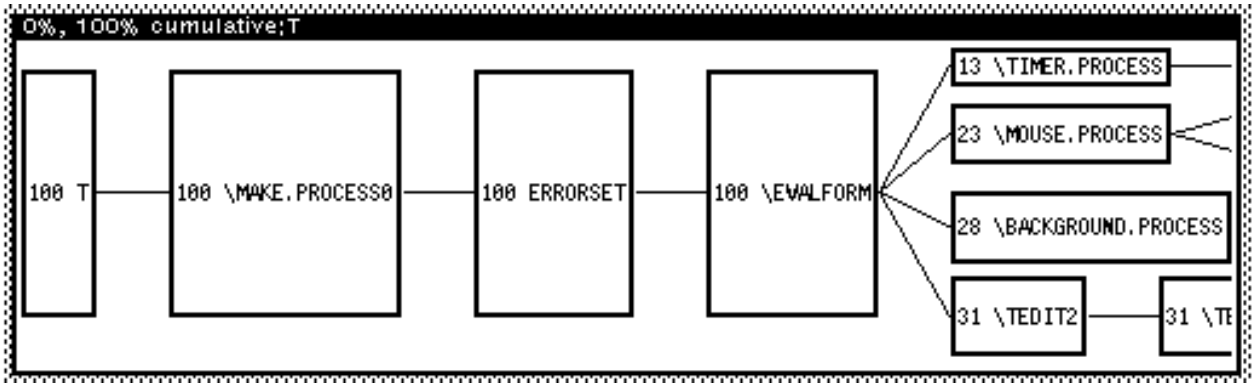
Calls (SPY.START), evaluates form, calls (SPY.END), and then returns the value of form. For example, (WITH.SPY (LOAD 'FOO)) is basically (PROGN (SPY.START) (PROG1 (LOAD 'FOO) (SPY.END]

To display the results, call the function SPY.TREE:

(SPY.TREE threshold individual mergetype depthlimit) [Function]

SPY.TREE displays the results of the last spy in a grapher window. There are a number of parameters that control the display, which you can either set when you call SPY.TREE, or set interactively with the menu. I normally just use (SPY.TREE), and use the menus. "threshold" is a number (defaults to 0), "individual" is either NIL or T, "mergetype" is one of (NONE, ALL, DEFAULT), and "depthlimit" is a number (defaults to NIL = arbitrary depth; not completely debugged for other values.)

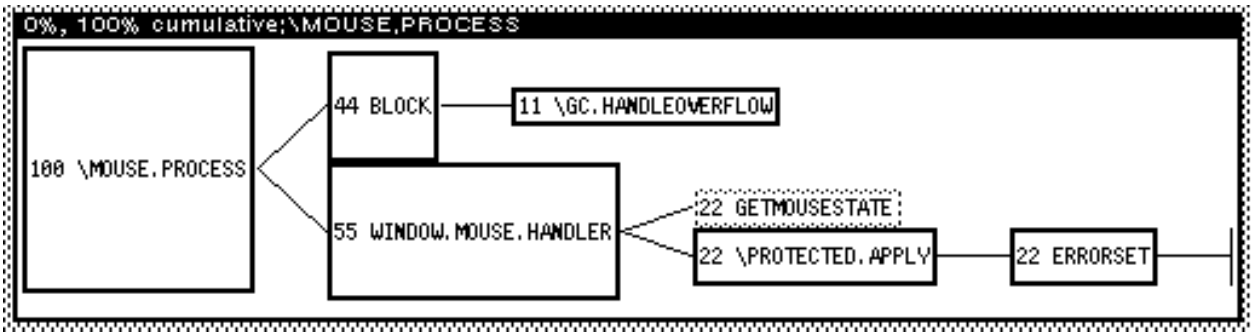
You will get a prompt to lay down a window, and then a graph will appear in it, something like this:



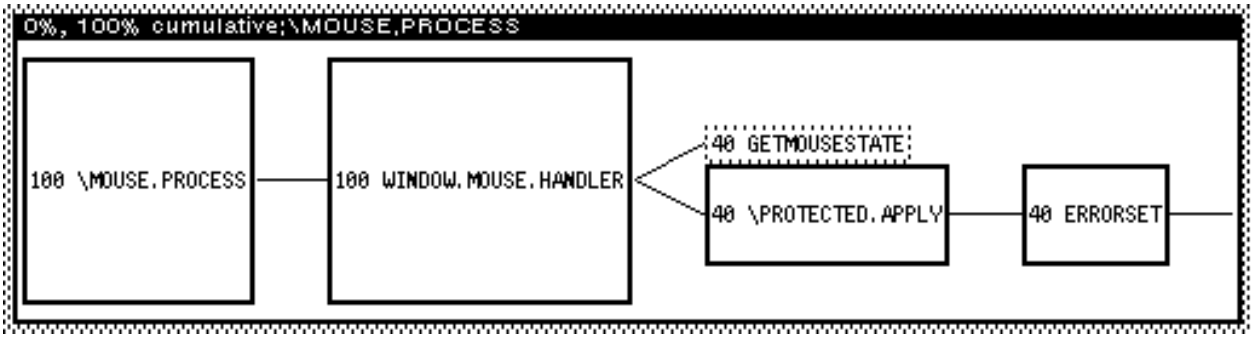
In this display, 100% of the time was spent under the top 4 frames that a process normally has, T, \MAKE.PROCESS0, ERRORSET, \EVALFORM. (SPY should probably eliminate those from the display, but it doesn't right now.). That time was then divided up among 4 processes \TIMER.PROCESS, \MOUSE.PROCESS, \BACKGROUND.PROCESS, \TEDIT2 The numbers to the left of the labels are the percentages. The height of the box is proportional to the percentage, (except that it is always made big enough to hold the label.) The width isn't significant - it is just wide enough to hold the name of the function.

If you left button any node, the window title will change to show the function name, and the individual and cumulative percentages, the first number individual total and the second is the cumulative. If you middle button the node, you will get a menu, with two items:

SubTree — Create another spy window which includes data only from this node and its descendents. Lets suppose that I was only interested in the actions that I had invoked via the mouse: I can middle-button \MOUSE.PROCESS and select the SubTree option. I then get a picture like this:



Delete Remove the selected from consideration in this window. For example, if you don't want to consider "BLOCK" in the graph at all, you can Delete the BLOCK box and get:



If you press the middle button while NOT on a node, you will get a menu that will let you change the paramters for this window:

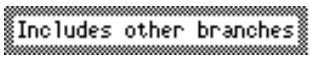
SetThreshold Sets the threshold for displaying a node. Any node whose percentage is below the threshold wont show up (unless it is needed to connect the graph together). You can set the initial threshold via the THRESHOLD argument to SPY.TREE; otherwise it defaults to 0.

Individual/Cumulative This is used to toggle between the display of Individual times and Cumulative. The initial default is Cumulative; you can set it to Individual by supplying T as the INDIVIDUAL argument to SPY.TREE (See below).

MergeNone/MergeAll/MergeDefault This controls merging of nodes. See discussion below.

MERGING

If two nodes are merged, then the merged node will include the time (and descendents) of the other. The display of a 'merged' node is different - it is shown with a thick gray border, e.g.



The time in a 'merged' node is the sum of the times for all occurrences. Other calls to the same function may show up as 'ghost' boxes, e.g.

Shown elsewhere

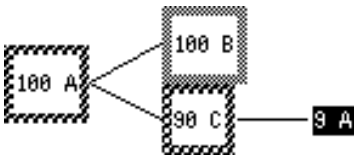
The case where a function is merged with a recursive call to itself is handled specially: the head of the recursion is marked with a wider checkered border

Head of recursive chain

and the tail of the recursion is inverted

End of recursive chain

In the recursive case, you can wind up with a situation like this:



In this case, A calls B, and calls C which calls A. All of the time is really spent in B, although only 10% is due to the call from the top-level, and 90% is under a call to C which called A which called B. In this situation C, is also recursive, of course, and also has the recursive border. If you find this display confusing, you should try the MergeNone option and see if you get a clearer picture.

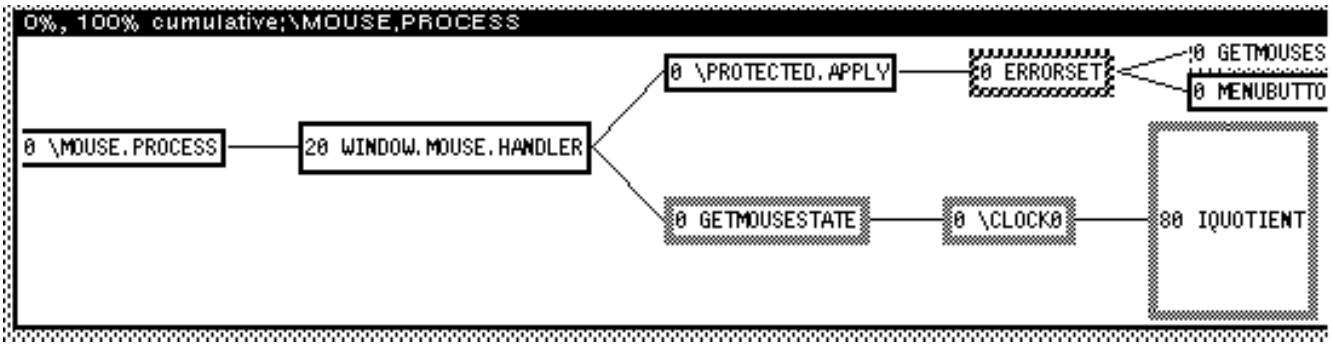
MergeDefault means to merge any function that is not in SPY.NOMERGEFNS, initially set to (ERRORSET \EVAL \EVALFORM APPLY EVAL).

MergeNone means not to merge at all
MergeAll means to merge any two nodes for the same function.

The default for Individual mode is MergeAll The default for Cumulative mode is MergeDefault.

"Individual" vs. "Cumulative"

SPY initially comes up with the height of the boxes showing the amount of time the function was on the current call stack. This is called "cumulative" mode, since the each function gets the time that both it and the functions it calls account for. There is another kind of display, called "Individual", in which the boxes are proportional to the amount of time the function was on the top of the stack. For example, if you change the \MOUSE.PROCESS display to Individual, you can get a display like this:



Which shows that 80% of the total time under \MOUSE.PROCESS was spent inside the function IQUOTIENT. (Whew, something going on wrong there. why isn't it getting the microcoded version of divide?) One thing to watch out for: when you switch between Individual and Cumulative modes, the "threshold" stays the same. (Sometimes the threshold for Individual needs to be higher; otherwise, functions will tend to disappear in the Individual tree. Also, switching to Individual also changes to "MergeAll", while switching to "Cumulative" changes you to "MergeDefault".

(SPY.LEGEND)
[function]
If you forget what all of the different shadings and borders mean, this function brings up a window which shows what they mean.

Internal controls:

SPY.FREQUENCY
[variable]
How many times per second to sample? Initially set to 10. (Maximum 60).

SPY.NOMERGEFNS
[variable]
Functions on this list won't get merged under MergeDefault. Includes (ERRORSET \EVALEVALFORM APPLY EVAL). You may need to add more.

SPY.TREE
[variable]
This variable (same name as the function) is used to hold the data from the last sampling. You can save it away and restore it using JGLYVARS.

SPY.BORDERS
[variable]
Used to control the border display on a tree. This is a list of (nodetype description borderwidth texture interiortexture)

(SPY.LEGEND)
[function]
Brings up a window which shows the interpretation of SPY.BORDERS.

SPY.FRAGMENTS
[variable]
If NIL, SPY will not include "ghost" boxes. Initially

SPY.SHOWCOUNTS
[variable]
If NIL, SPY will not display the percentage in the node-box unless the node-box has been made higher to accomodate the text in the node. Initially T.

SPY.FONT
[variable]
Font used to display node labels. Initially (GACHA 10).

SPY.MAXLINES [variable]
Maximum height of a node in the graph, measured in multiples of the font height of SPY.FONT.

KNOWN PROBLEMS WITH THIS VERSION OF SPY

The Dolphin and Dorado have a bug which will cause incorrectly large amounts of time to be accumulated to the routine which get and send packets to the 3MB Ethernet. It will show up as samples under \3MBGETPACKET (or whatever). The rest of the data in the SPY display is relatively accurate, and you may need to 'delete' that node.

SPY doesn't know anything about the interpreter or the internal workings of Interlisp-D. Internal functions which are not "real frames" and don't normally show up on BT backtraces (but do on BT!) will be shown in spy. This includes things like \INTERPRETER, which will appear underneath any interpreted function call.

Finally, a number of enhancements and extentions are planned; suggestions and requests are welcome.