
Subject: Re: Garbage collection and Memory
Posted by [landers](#) on Fri, 03 Jun 1994 18:08:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <1994Jun2.155555@estwm0.wm.estec.esa.nl>, hevans@estwm0.wm.estec.esa.nl (Hugh Evans) writes:

|> Hello,

|>

|> I am using PV~Wave v4.01 (VAX/VMS) and after loading in large data sets and
|> manipulating them, PV~Wave runs out of memory (core). This is despite creating
|> more variables.

|>

|> The question I have is: Is there any command that invokes a garbage collector
|> to clean up the memory used? Or do I just have to save the session, exit and
|> restart the session?

That's one way. Or you can DELVAR all the unused variables. Or SAVE what you want to keep, then .RNEW a something.pro (this cleans all variables - unfortunately there's no DELVAR,/All), and then RESTORE. This will re-pack your variables into memory, and free up the unused stuff.

Another thing to do is be more careful with memory allocation. Be sure to free memory when you're done with it. Things like this are helpful:

```
a = 0 ; free up any memory associated with a (assuming it's an array)
a = some_large_array_expression
; some code...
a = 0 ; done with a, so free it up
```

When you do an assignment (like the 2nd expression above), WAVE has to allocate memory for the intermediate answer (the right hand side), and then do the assignment (point the left hand side variable at the result). If the left hand side is already defined, then it's memory is not free'd until just before the assignment. So you can free up most of the memory by doing an 'a = 0' so there's memory available for the RHS evaluation.

Of course this doesn't always cure everything, because WAVE has to be able to find enough unbroken memory for your results. Eventually, (depending on your code), everything can get fragmented (just like VMS does to your disks).

A really good way to frag up your memory is by 'growing' arrays. Like this:

```
while ( condition ) do begin
  x = calcuation()
  a = [ a, x ]
end
```

Each time this loops thru, it needs a piece of memory a little bigger than the

last time. It probably has to keep allocating a new block of memory for each loop, except for a few times when you'll get lucky. You use something like $(1 + 2 + 3 + \dots + N)$ hunks of memory, rather than just (N) .

To fix this, allocate the array before the loop, and keep track of a pointer to the last used piece. Then trim the array when you're done. Either allocate the array to as much as you'll need (or more), or put some code in to grow it by a large amount (like double it) when necessary.

```
a = fltarr( largest_possible )
i = 0
while condition do begin
  x = calc()
  a(i) = x
  i = i + n_elements(x)
end
a = a(0:i-1)
```

Remember that once you start a WAVE session, you can never 'give memory back' to the operating system. It just marks it as unused, and will reuse it. this is an artifact of C's malloc() and free() procedures.

```
|> Regards,
|> --
|> Hugh Evans
|> European Space Research and Technology Centre - Noorwijk, Netherlands
|> Internet: hevans@wm.estec.esa.nl SPAN: ESTWM2::hevans
|>
|> But she stopped herself. You didn't juggle matches in a firework factory.
|> (Terry Pratchett, Witches Abroad)
```

;Dave
