Subject: Re: % FINDFILE: Error managing child process.
Posted by George N. White III on Mon, 26 Jul 2004 14:08:12 GMT
View Forum Message <> Reply to Message

On Sun, 25 Jul 2004, Timm Weitkamp wrote:

> On 24.07.04 at 21:21 -0500, Craig Markwardt wrote:
>
>> Timm Weitkamp <dont.try@this.address> writes:
>>> I am having an annoying problem with FINDFILE on a Sunblade with IDL 6.0
>>> (version = { sparc sunos unix Solaris 6.0 Jun 27 2003 64 64}). I run an
>>> IDL program that runs fine on other machines (PCs running Linux) and used
>>> to work on this Sunblade too, and now it stops with an error message at a
>>> call to FINDFILE. The call is alright, the problem is that FINDFILE seems
>>> to spawn an "ls" or something, because I get the same error when I spawn
>>> "ls" by hand:
>>>
>>> IDL> $ls
>>> % Error managing child process.
>>> Not enough space
>>>
>>> What is this? Bug? Feature? Memory problem? It actually occurs with *any*
>>> spawned command (I tried "cat" or even non-existent commands with the same
>>> result). Can I circumvent this somehow? Any help appreciated. Here's the
>>> system info:
>>
>> I seem to recall a friend having a problem like this. Somehow, by
>> accident, his "/tmp" scratch area was non-writeable, except by root.
>> Once we made it writeable, then all sorts of things started to work
>> again. We speculated that somehow IDL was attempting to make a
>> temporary file, which it couldn't do when permissions were wrong.
>
> Thank you for this hint, Craig, which I will keep in mind. At first
> glance, though, it doesn't exactly seem to be that (/tmp/ is writable).
> Another thing is that I now tried with different IDL releases (5.4, 5.6,
> and 6.0). The problem occurs with 5.6 and 6.0, but not with 5.4.
> Unfortunately my code relies on some functionality introduced in 5.6.
>
> Also, it only occurs after allocating a considerable bunch of memory for
> an array (something like 4 GB out of the total of 6 GB of RAM available).
> But once the bug strikes, even freeing memory with DELVAR or TEMPORARY
> does not help anymore.

The "real" unix fork of a large process requires reserving double the
memory. The IRIX documentation says:

"The IRIX system allows programs occupying more space than the system
limit to run, since each program is only partially loaded into memory

at any given time. One of the effects of this policy is that the IRIX system has to preallocate swap space based on likely future usage, and sometimes this prediction is incorrect. When the swap space is actually needed, the IRIX system allocates the most convenient available space, not the specific space allocated. So the physical allocation is separate from the accounting allocation.

If your system preallocates all your swap space, but the space has not yet been used, it may appear that your system is running out of swap space when it is not. It is possible that your system has simply preallocated the rights to future swap space to existing processes, and no new processes can allocate space due to the strict swap space accounting in the IRIX system."

To work around this, it is customary to add "virtual" swap space.

> I worked on other Sunblades before (albeit with more memory available) and
> never ran into this. May have to contact RSI.

RSI's documentation says:

"RSI strongly recommends the use of the FILE_SEARCH function in place of the FINDFILE function. FILE_SEARCH is more platform-independent, provides greater functionality, and is easier to use than FINDFILE. FILE_SEARCH is ultimately intended as a replacement for FINDFILE."

--
George N. White III <aa056@chebucto.ns.ca>
Head of St. Margarets Bay, Nova Scotia, Canada