Subject: Re: Interrupted System Calls reading from NFS on Sun Solaris
Posted by caron on Thu, 30 Jun 1994 16:32:13 GMT
View Forum Message <> Reply to Message

I know that interrupted system calls in UNIX are a normal occurence. The
correct thing to do is (almost always) to simply re-issue the call.
(See Leffler et al, p 47-48). What is surprising is that this behavior
is seen at this high of an API. Typically you are using some library routine
that takes care of it (e.g. fread()). What is READ_FLOATS actually calling?
If it is calling an i/o routine that does not handle return EINTR, (read() ?)
then I would guess that the correct solution is that READ_FLOATS should
handle it (by looking for it and then reissuing read), and not
allow it to propagate up.

I am not sure on the details of how NFS works, but EINTR is gotten when the
system issues a system call that may take a long time to complete (eg I/O)
and a signal comes along that must be serviced. I think that "saving the state"
of the system call is too hard, so after servicing the signal, the OS
just returns from the system call with EINTR, more or less telling the
calling routine to "try again". Probably theres lots more subtleties, but I
think thats the vanilla explanation.