Subject: Re: Non-Blocking I/O
Posted by thompson on Thu, 11 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

ashmall@my-dejanews.com (Justin Ashmall) writes:

> Just a thought, but would an FSTAT on the unit number give you any information
> as to whether there was data waiting to be read?

> Justin

Probably not. I've dealt with situations where we've had to read from a file which was open for write by another process. As I recall, the behavior of FSTAT was somewhat flakey under those conditions.

The situation we were dealing with was to read an incoming spacecraft telemetry stream. Since there already was a process (written in C) which was archiving the telemetry stream into data files, what we ended up doing was to simply read those files while they were still being written. That way, we avoided the whole pipe/fifo business. Sounds like that wouldn't help you, though.

Our original scheme was to use a two-way socket connection between IDL and a C process which was handling telemetry reception. IDL would send out a request for data to the socket, and the C process would either respond with a packet, or with a "no-data-yet" message. That way, IDL would always read back something.

> In article <36C2E662.F81D5072@Physik.Uni-Marburg.De>, Ruediger Kupper
> <Ruediger.Kupper@Physik.Uni-Marburg.De> wrote:
>> Hi!
>>
>> This is a question regarding Inter Process Communication
>> (IPC) in a UNIX environment:
>>
>> Is there any way to tell IDL not to wait for the next
>> incoming data when reading from a file?
>> Attempting to read from an IPC channel (a pipe or fifo)
>> using READF or READU will cause IDL to hang until this read
>> attempt is successful. (Pipes or fifos do not produce an
>> EOF-signal unless they are explicitely closed by all sending
>> processes.) This blocking behaviour is undesired if you need
>> to check more than one IPC channel for any waiting data, or
>> if you want to incorporate such a check into an event loop.
>>
>> The only solution I can think of is to provide support for
>> non-blocking I/O by a set of C-routines which could be
>> linked to IDL via CALL_EXTAERNAL, but I would prefer using

>> any "pure IDL" concept.
>>
>> If anyone out there ran into the same problem, this person
>> could make a poor, frustrated IDL-programmer very happy by
>> posting me a little hint...
>>
>> Best regards,
>> Ruediger.
>>
>>
