Subject: : AmigaOS4
Topic: : Memory protection and tasks/processes
Re: Memory protection and tasks/processes
Author: : LiveForIt
Date: : 2021/3/2 16:52:44
URL:

@afxgroup

is crashing in the kernel with a DSI and freeze the entire OS. Is this normal?

YES..

anyway this looks like nonsense.
“Delete” is not the same as “Release”
Delete is like free or delete object,
while release in this context is unclocking the dirent_lock, so another task can use it, at least on (Windows and AmigaOS)

If the code looks like:

```
 if (a == NULL) goto cleanup;    // will try release lock that is not obtained, and crash maybe.
ObationShampre(dirent_lock)

if (b == NULL) goto cleanup;    // will try release lock that is obtained.
ReleaseSemaphore(dirent_lock);

cleanup:
__delete_semaphore(dirent_lock);    // will unlock it, but not delete/free it.
```

in this case you don’t know if you lock or not.

You have case like

O = Obtain
R = Release
W = write
r = Read
F = Free mem
A = Alloc Mem
. = time tick.

0 ms time -------- > Lots of ms Time

```
Task1 ....O..wwww.R............O..rrr.F.....A.wwwwwww.R.
Task2 ..............O...www...R......R.................
Task3 ................................O..rrrr..R.....
```

in this case Task3 thinks it can read but because there is a bug, its released the lock twice in task2.
Task3 will read freed/corrupted data, and crash with DSI.

Or you can have cases where…
Task1 ….. O ….. O [freeez] [then crash…]

Task1 freeze becouse Semaphore is already obtained.

way it crashes is, because the message port gets new messages that is never replyed, and so never deleted, so message port fills up until system crashes. (most likely)

or/and the sender maybe locks up becouse it waiting to replay.