

Re: How to allocate memory for a linked list of pointers in a kernel process

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.kernel/2008-02/msg00127.htm>

- *From:* Tim Roberts <timr@xxxxxxxxxx>
 - *Date:* Mon, 11 Feb 2008 23:11:58 -0800
-

joemango <joseandremorales@xxxxxxxxxx> wrote:

Hi everyone, i am building a short program that builds several linked lists using pointers of course. Its my first time doing this in the kernel, i need to know the best way to allocate memory when i create a new pointer that will for multiple pointers to be created without causing a system crash, I had started with pointer delclaration getting memory allocation with the ExAllocatePoolXxx function. but quickly found that would not work i was declaring too many pointers and used up all of the small amount of pool memory causing a BSOD !

For gosh sakes, how many pointers are you allocating? If you run with Driver Verifier, ExAllocatePool ends up allocating a page per allocation, which could eat up space, but otherwise I'm confused.

so now i need to know what i can use to access memory for my linked lists that wont cause teh BSOD and the right way to do to be efficient with memory usage.

Linked lists of what? Usually, the linked list pointers will be part of another, larger structure. When the kernel routines need to allocate a bunch of items of the same size, they use a lookaside list (ExInitializePagedLookasideList, ExAllocateFromPagedLookasideList, etc).

Are you aware that the kernel contains routines to do doubly linked lists (InitializeListHead, InsertTailList, InsertHeadList, etc)?

—
Tim Roberts, timr@xxxxxxxxxx
Providenza & Boekelheide, Inc.