

```

/*
THRASH.C -- Illustrates virtual memory in Windows Enhanced mode
from December 1992 Microsoft Systems Journal
Andrew Schulman
*/

#include <stdlib.h>
#include <stdarg.h>
#include <string.h>
#include <time.h>
#include "windows.h"

#define ALLOC_SIZE 4096
#define MAX_BLOCK 4096

typedef char far *FP;
static FP matrix[MAX_BLOCK] = {0};

int wprintf(char *fmt, ...)
{
    char *buf = malloc(1024); // local alloc
    int ret;
    va_list marker;
    va_start(marker, format);
    ret = wvsprintf(buf, format, marker);
    MessageBox(0, buf, "TEST", MB_OK);
    free(buf);
    return ret;
}

#define OK(msg) (MessageBox(0, msg, "THRASH", MB_OKCANCEL) == IDOK)

int PASCAL WinMain(HANDLE hInstance, HANDLE hPrevInstance,
    LPSTR lpCmdLine, int nCmdShow)
{
    int blocks = 0;
    unsigned long bytes = 0;
    time_t t1, t2;
    HANDLE h;
    int i, j;

    memset(matrix, 0, MAX_BLOCK * sizeof(char far *));

    time(&t1);
    while (h = GlobalAlloc(GMEM_MOVEABLE | GMEM_ZEROINIT, ALLOC_SIZE))
    {
        char far *fp = GlobalLock(h);
        if (blocks < MAX_BLOCK)
            matrix[blocks] = fp;
        _fmemset(fp, 'x', ALLOC_SIZE); // touch every byte
        blocks++;
        bytes += ALLOC_SIZE;
        Yield(); // let other apps run
    }
    time(&t2);
}

```

```

wprintf("Allocated %lu bytes in %u blocks\nin %lu seconds",
        bytes, blocks, t2 - t1);

if (! OK("Starting row-major access"))
    return 1;
time(&t1);
// don't Yield in here
for (i=0; i<blocks; i++)
    for (j=0; j<ALLOC_SIZE; j++)
        matrix[i][j] = 'y';
time(&t2);
wprintf("Row-major access in %lu seconds", t2-t1);

if (! OK("Starting (possibly thrashing) column-major access"))
    return 1;
time(&t1);
// don't Yield in here
for (j=0; j<ALLOC_SIZE; j++)
    for (i=0; i<blocks; i++)
        matrix[i][j] = 'y';
time(&t2);
// realistically, probably run out of patience and reboot
// machine somewhere above if Enhanced mode and PAGING=ON
wprintf("Column-major access in %lu seconds", t2-t1);

return 0;                // release all memory
}

```