

```
/*  
SHOWPHYS.C -- Display page-table entries for a given segment  
from December 1992 Microsoft Systems Journal  
Andrew Schulman  
*/
```

```
#include <stdlib.h>  
#include <stdio.h> // for sscanf  
#include <string.h>  
#include <dos.h>  
#include "windows.h"  

```

```
void fail(char *s)  
{  
    MessageBox(NULL, s, "SHOWPHYS", MB_OK);  
    exit(1);  
}
```

```
static char buf[2048] = {0};
```

```
int PASCAL WinMain(HANDLE hInstance, HANDLE hPrevInstance,  
    LPSTR lpCmdLine, int nCmdShow)
```

```
{  
    char tmp[128];  
    char title[64];  
    WORD far *pagedir;  
    DWORD far *pagetab;  
    DESCRIPTOR far *desc;  
    SELECTOR sel;  
    DWORD base, limit, offset, b;  
    WORD seg, pagetab_num, page;  
    BOOL first_page;
```

```
    if (!(GetWinFlags() & WF_ENHANCED))  
        fail("This program requires Windows Enhanced mode");
```

```
    pagedir = get_pagedir();
```

```
    lstrcpy(tmp, lpCmdLine);  
    sscanf(tmp, "%04X", &seg);
```

```
    for (;;) 
```

```
{  
    #if 1  
        sel = *((SELECTOR *) &seg);  
        if (sel.ti == 0) // GDT: (seg & 4) == 0  
        {  
            get_gdt();  
            if ((seg & ~8) > GetSelectorLimit(FP_SEG(gdt)))  
                fail("Invalid selector");  
            desc = &gdt[sel.index]; // seg >> 3  
        }  
        else // LDT: (seg & 4) == 1  
        {  
            DESCRIPTOR far *ldt;
```

```

    ldt = get_ldt();
    if ((seg & ~8) > GetSelectorLimit(FP_SEG(ldt)))
        fail("Invalid selector");
    desc = &ldt[sel.index]; // seg >> 3
}
base = get_base(desc);
limit = get_limit(desc);
#else
// these don't know about GDT selectors
base = GetSelectorBase(seg);
limit = GetSelectorLimit(seg);
#endif

// MessageBox can only display about 18 lines
if (limit > (18L * 4096L))
    fail("Segment too large");

wsprintf(title, "SHOWPHYS %04X - Base %lX - Limit %lX",
    seg, base, limit);

offset = base & 0xFFF; // offset within first page
limit += offset; // how many pages to show
base &= ~0xFFFL; // turn off bottom 4k

strcpy(buf, "Dir Page\tPhysical\n");

for (b=base, first_page=TRUE; b<base+limit; b+=4096) // for each page
{
    // should use bitfields
    pagetab_num = (b & 0xFFC00000L) >> 22;
    page = (b & 0x3FF000L) >> 12;

    pagetab = get_pagetab(pagedir, pagetab_num);

    wsprintf(tmp, "%03X %03X\t\t", pagetab_num, page);
    strcat(buf, tmp);

    // check present bit
    if (pagetab[page] & 1)
        wsprintf(tmp, "%08lX", pagetab[page] & 0xFFFFF000L);
    else
        wsprintf(tmp, "not present");
    strcat(buf, tmp);

    // if doesn't start on page boundary
    if (first_page && offset)
    {
        wsprintf(tmp, " + %03X", offset);
        strcat(buf, tmp);
        first_page = FALSE;
    }

    strcat(buf, "\n");

    free_mapped_linear(pagetab);
}

```

```
    }  
    if (MessageBox(NULL, buf, title, MB_OKCANCEL) == IDCANCEL)  
        break;  
    buf[0] = '\0'; // start over again  
}  
  
free_mapped_linear(pagedir);  
return 0;  
}
```