突破DOS实模式限制直接访问4GB内存 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/454/2021_2022__E7_AA_81_E7_A0_B4DOS_E5_c98_454800.htm PM_Service: Mov AX,16 Mov _seg,AX Mov EBX,CR0 And EBX,0fffffffeh Mov CR0,EBX DB 0eah DW Real_Service DW seg Real_Service Real_Service: Lgdt FWORD Ptr [Old_GDTR] Popfd .恢复现场 Popad Pop ES Pop DS Jmp _Exit MyGdt DQ 0 DW -1,0,9a00h,0 DW -1,0,9200h,0cfh DQ 0 Old_GDTR DW 0,0,0 GDTR DW 0,0,0 _Exit: Endm 在这里为了方便我只把FS改成4GB段，读者可以按需要自行决定使用哪个段寄存器。只要将这段代码拷贝到你的程序中，然后在开始的时候调用它，就可以通过该段寄存器直接访问大内存了，爽吧！ 最后还有一点一定要注意：如果你的程序运行时有任何扩展内存管理程序存在（HIMEM、EMM386等）都要千万小心，因为很容易会破坏到它们的内部数据或其他程序的数据，如果是这样就只有死机一条路可走了。切记切记！我的建议是最好从内存顶端开始使用扩展内存。这时破坏其他数据的可能要小一些。local MyGdt,PM_Service,Old_GDTR,GDTR,Real_Service,MyGdt local _Exit Push DS Push ES Pushad Pushfd .保护现场 Sub EBX,EBX Mov BX,CS Mov DS,BX Shl EBX,4 Push EBX Rol EBX,8 Mov BYTE Ptr MyGdt[8 7],BL Mov BL,BYTE Ptr MyGdt[8 5] Ror EBX,8 Mov DWORD Ptr MyGdt[8 2],EBX Pop EBX lea EBX,[EBX MyGdt] Mov DWORD Ptr [GDTR 2],EBX Mov WORD Ptr [GDTR],31 .建立新的GDTR Cli Sgdt FWORD Ptr [Old_GDTR] .保存旧的GDTR Lgdt FWORD Ptr [GDTR] .设置新的GDTR

Mov EBX,CR0 Or BL,1 Mov CR0,EBX .进入保护模式 DB 0eah DW PM_Service DW 8 .跳转到保护模式代码执行|PM_Service: Mov AX,16 Mov _seg,AX Mov EBX,CR0 And EBX,0fffffffeh Mov CR0,EBX DB 0eah DW Real_Service DW seg Real_Service Real_Service: Lgdt FWORD Ptr [Old_GDTR] Popfd .恢复现场 Popad Pop ES Pop DS Jmp _Exit MyGdt DQ 0 DW -1,0,9a00h,0 DW -1,0,9200h,0cfh DQ 0 Old_GDTR DW 0,0,0 GDTR DW 0,0,0 _Exit: Endm 在这里为了方便我只把FS改成4GB段，读者可以按需要自行决定使用哪个段寄存器。只要将这段代码拷贝到你的程序中，然后在开始的时候调用它，就可以通过该段寄存器直接访问大内存了，爽吧！最后还有一点一定要注意：如果你的程序运行时有任何扩展内存管理程序存在（HIMEM、EMM386等）都要千万小心，因为很容易会破坏到它们的内部数据或其他程序的数据，如果是这样就只有死机一条路可走了。切记切记！我的建议是最好从内存顶端开始使用扩展内存。这时破坏其他数据的可能要小一些。