

OS lab 5

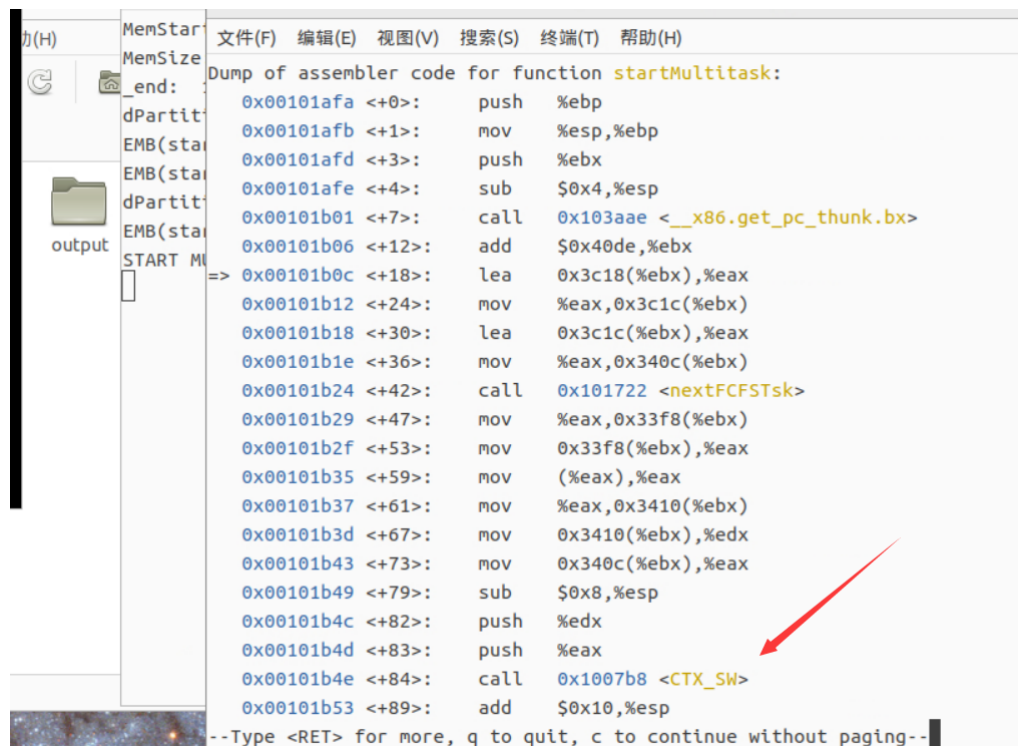
PB21051012 刘祥辉

思考题

- 在上下文切换的现场维护中, pushf和popf对应, pusha和popa对应, call和ret对应, 但是为什么 CTS_SW 函数

中只有ret而没有call呢

call 指令用于调用函数或跳转到一个子程序(函数)的代码段, 并在子程序执行完毕后返回到调用它的地方。ret 指令用于将控制权从被调用函数返回到调用函数, CTS_SW函数是被调用函数, 在task.c文件中有代码行 CTS_SW(prevTSK_StackPtr,nextTSK_StackPtr); 查看汇编代码有:



```
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)
Dump of assembler code for function startMultitask:
0x00101afa <+0>:  push  %ebp
0x00101afb <+1>:  mov   %esp,%ebp
0x00101afd <+3>:  push  %ebx
0x00101afe <+4>:  sub   $0x4,%esp
0x00101b01 <+7>:  call  0x103aae <__x86.get_pc_thunk.bx>
0x00101b06 <+12>: add    $0x40de,%ebx
=> 0x00101b0c <+18>: lea    0x3c18(%ebx),%eax
0x00101b12 <+24>: mov    %eax,0x3c1c(%ebx)
0x00101b18 <+30>: lea    0x3c1c(%ebx),%eax
0x00101b1e <+36>: mov    %eax,0x340c(%ebx)
0x00101b24 <+42>: call   0x101722 <nextFCFSTsk>
0x00101b29 <+47>: mov    %eax,0x33f8(%ebx)
0x00101b2f <+53>: mov    0x33f8(%ebx),%eax
0x00101b35 <+59>: mov    (%eax),%eax
0x00101b37 <+61>: mov    %eax,0x3410(%ebx)
0x00101b3d <+67>: mov    0x3410(%ebx),%edx
0x00101b43 <+73>: mov    0x340c(%ebx),%eax
0x00101b49 <+79>: sub    $0x8,%esp
0x00101b4c <+82>: push   %edx
0x00101b4d <+83>: push   %eax
0x00101b4e <+84>: call   0x1007b8 <CTS_SW>
0x00101b53 <+89>: add    $0x10,%esp
--Type <RET> for more, q to quit, c to continue without paging--
```

故与ret配对的call在task.c中。

- 谈一谈你对 stack_init 函数的理解。

stack_init初始化栈, 在CTS_S函数中确保任务在启动时能够正确地加载寄存器和堆栈指针, 并开始执行指定的函数。

- `*(*stk)-- = (unsigned long) 0x08;`: 将 0x08 存储到堆栈帧中, 作为高地址部分。这通常用于存储 CS 寄存器的值, 它指示代码段的选择子。
- `*(*stk)-- = (unsigned long) task;`: 将 task 函数的地址存储到堆栈帧中, 作为 EIP 寄存器的值。EIP 寄存器存储着指令指针, 即下一条将要执行的指令的地址。
- `*(*stk)-- = (unsigned long) 0x0202;`: 将 0x0202 存储到堆栈帧中, 作为 FLAG 寄存器的值。FLAG 寄存器中的标志位用于控制和影响程序的运行状态, 例如进位标志、零标志、符号标志等。

其他寄存器存入的值可能是为了方便调试。

- myTCB结构体定义中的stack[STACK_SIZE]的作用是什么？BspContextBase[STACK_SIZE]的作用又是什么？

myTCB中的stack[STACK_SIZE]是为了初始化寄存器，在上下文切换中能够正确返回到相应的函数中
BspContextBase[STACK_SIZE]是存储被切换处的任务的堆栈指针，以供下次切换回来恢复寄存器值。

- prevTSK_StackPtr是一级指针还是二级指针？为什么？

二级指针，BspContext是指向unsigned long数据类型的指针，prevTSK_StackPtr是指向BspContext的指针，所以是二级指针。

运行结果

