Nama : Agus Satrio Rukmana

NIM : L200170117

1. Masuk ke OS dengan perintah ‘CD OS’, lalu dilanjutkan dengan mamasukan perintah ‘setpath’

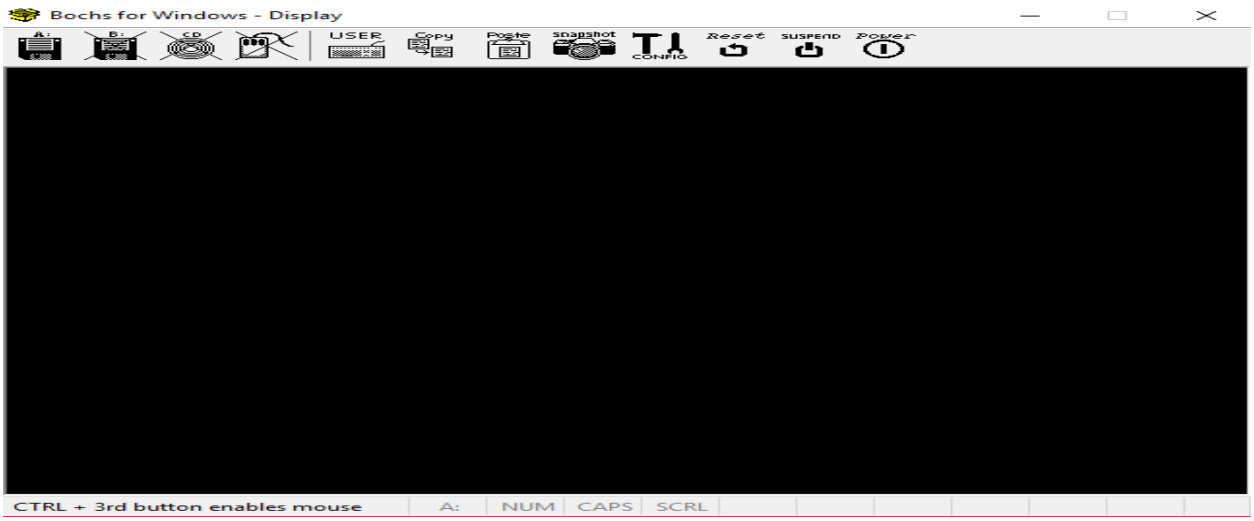
```
C:\>cd os

C:\OS>setpath

C:\OS>Path=C:\OS\Dev-Cpp\bin;C:\OS\Bochs-2.3.5;c:\OS\Perl;C:\Windows;C:\Windows\System32
C:\OS>cd lab/lab3
```

2. Selanjutnya ketikan perintah ‘type s.bat’

```
C:\OS\LAB\LAB3>type s.bat
..\..\bochs-2.3.5\bochsdbg -q -f bochsrc.bxrc
```

3. Dilanjutkan dengan memasukan perintah ‘s’ untuk masuk ke PC simulator dan akan muncul seperti gambar dibawah ini

```
C:\OS\LAB\LAB3>s

C:\OS\LAB\LAB3>..\..\bochs-2.3.5\bochsdbg -q -f bochsrc.bxrc
00000000000i[APIC?] local apic in  initializing
========================================================================
                        Bochs x86 Emulator 2.3.5
            Build from CVS snapshot, on September 16, 2007
========================================================================
00000000000i[      ] reading configuration from bochsrc.bxrc
00000000000i[      ] installing win32 module as the Bochs GUI
00000000000i[      ] using log file bochs.log
Next at t=0
(0) [0xfffffff0] f000:fff0 (unk. ctxt): jmp far f000:e05b         ; ea5be000f0
```

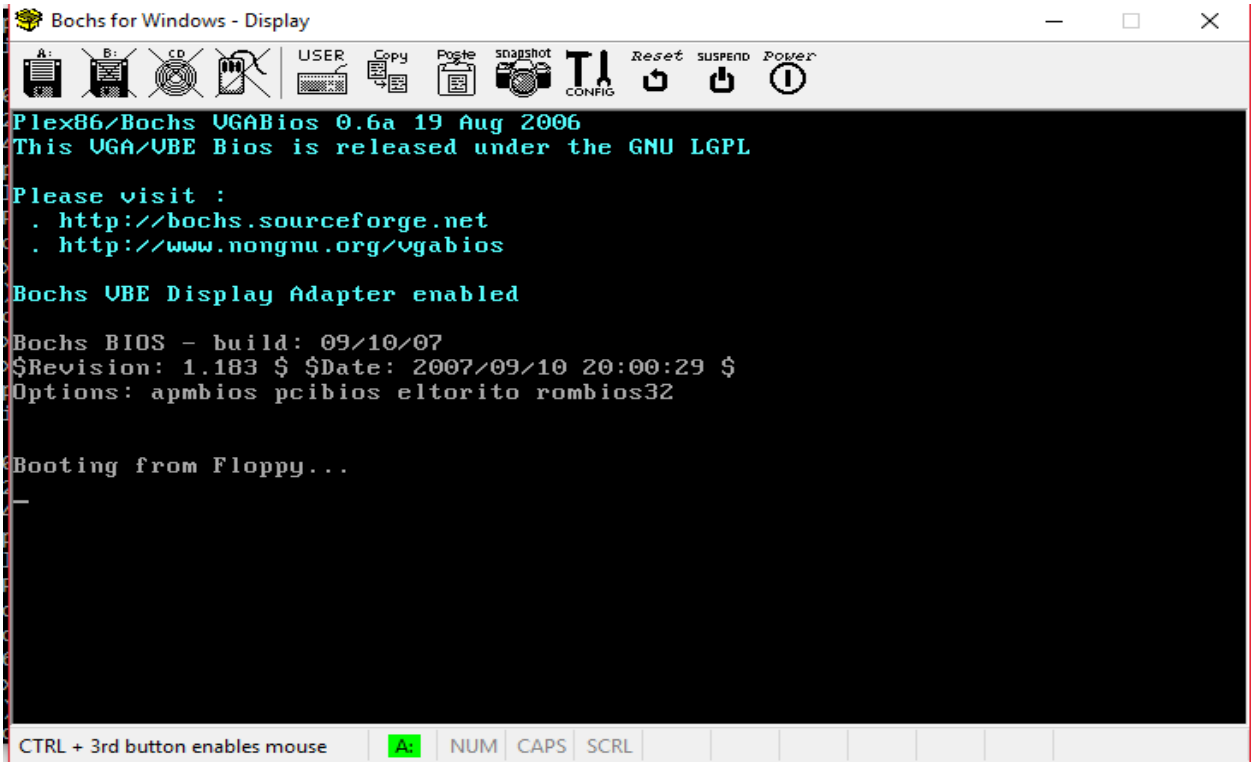


4. Ketikan perintah ‘r’ hingga muncul seperti gambar dibawah ini

```
<bochs:1> r
rax: 0x00000000:00000000 rcx: 0x00000000:00000000
rdx: 0x00000000:00000f20 rbx: 0x00000000:00000000
rsp: 0x00000000:00000000 rbp: 0x00000000:00000000
rsi: 0x00000000:00000000 rdi: 0x00000000:00000000
r8 : 0x00000000:00000000 r9 : 0x00000000:00000000
r10: 0x00000000:00000000 r11: 0x00000000:00000000
r12: 0x00000000:00000000 r13: 0x00000000:00000000
r14: 0x00000000:00000000 r15: 0x00000000:00000000
rip: 0x00000000:0000fff0
eflags 0x00000002
IOPL=0 id vip vif ac vm rf nt of df if tf sf zf af pf cf
```

5. Dilanjutkan dengan memasukan perintah ‘s’

```
<bochs:2> s
Next at t=1
(0) [0x000fe05b] f000:e05b (unk. ctxt): xor ax, ax                ; 31c0
<bochs:3> r
rax: 0x00000000:00000000 rcx: 0x00000000:00000000
rdx: 0x00000000:00000f20 rbx: 0x00000000:00000000
rsp: 0x00000000:00000000 rbp: 0x00000000:00000000
rsi: 0x00000000:00000000 rdi: 0x00000000:00000000
r8 : 0x00000000:00000000 r9 : 0x00000000:00000000
```

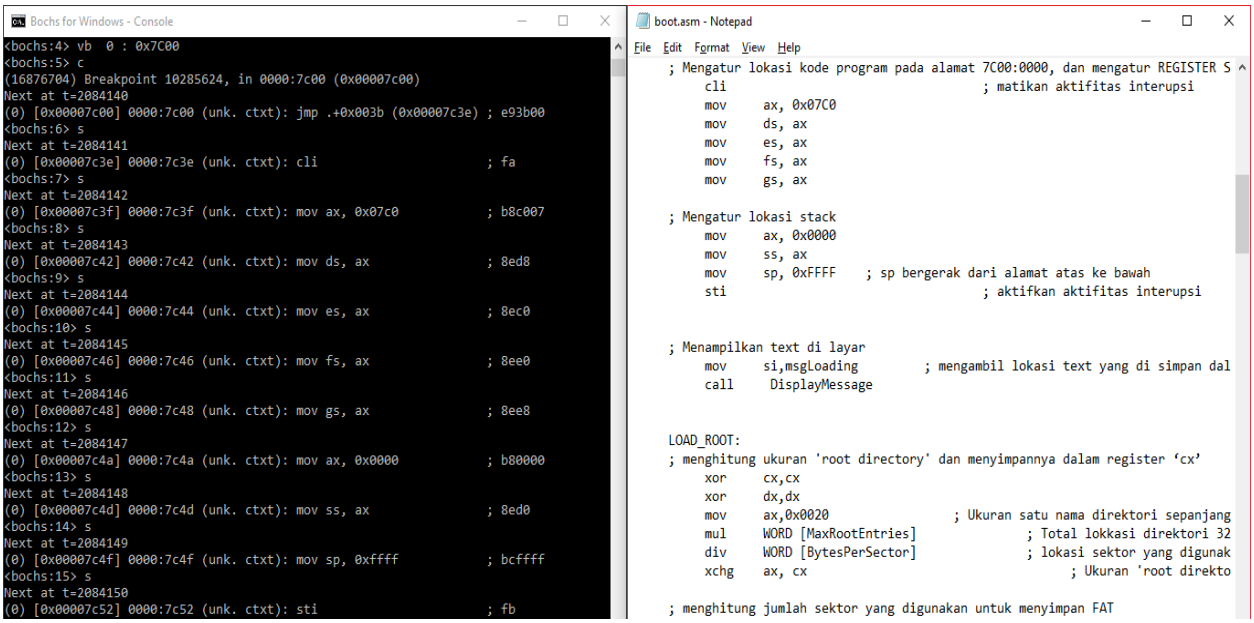



```
eflags 0x00000002
IOPL=0 id vip vif ac vm rf nt of df if tf sf zf af pf cf
<bochs:4> vb  0 : 0x7C00
<bochs:5> c
```

6. Ketikan perintah ‘q’ untuk menghentikan proses debugging

```
<bochs:18> q
# In bx_win32_gui_c::exit(void)!

Bochs is exiting. Press ENTER when you're ready to close this window.
```

7. Samakan proses debugging dengan file boot.asm



Bochs for Windows - Console

```
<bochs:4> vb  0 : 0x7C00
<bochs:5> c
(16876704) Breakpoint 10285624, in 0000:7c00 (0x00007c00)
Next at t=2084140
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:6> s
Next at t=2084141
(0) [0x00007c3e] 0000:7c3e (unk. ctxt): cli                       ; fa
<bochs:7> s
Next at t=2084142
(0) [0x00007c3f] 0000:7c3f (unk. ctxt): mov ax, 0x07c0            ; b8c007
<bochs:8> s
Next at t=2084143
(0) [0x00007c42] 0000:7c42 (unk. ctxt): mov ds, ax                ; 8ed8
<bochs:9> s
Next at t=2084144
(0) [0x00007c44] 0000:7c44 (unk. ctxt): mov es, ax                ; 8ec0
<bochs:10> s
Next at t=2084145
(0) [0x00007c46] 0000:7c46 (unk. ctxt): mov fs, ax                ; 8ee0
<bochs:11> s
Next at t=2084146
(0) [0x00007c48] 0000:7c48 (unk. ctxt): mov gs, ax                ; 8ee8
<bochs:12> s
Next at t=2084147
(0) [0x00007c4a] 0000:7c4a (unk. ctxt): mov ax, 0x0000            ; b80000
<bochs:13> s
Next at t=2084148
(0) [0x00007c4d] 0000:7c4d (unk. ctxt): mov ss, ax                ; 8ed0
<bochs:14> s
Next at t=2084149
(0) [0x00007c4f] 0000:7c4f (unk. ctxt): mov sp, 0xffff            ; bcffff
<bochs:15> s
Next at t=2084150
(0) [0x00007c52] 0000:7c52 (unk. ctxt): sti                       ; fb
```

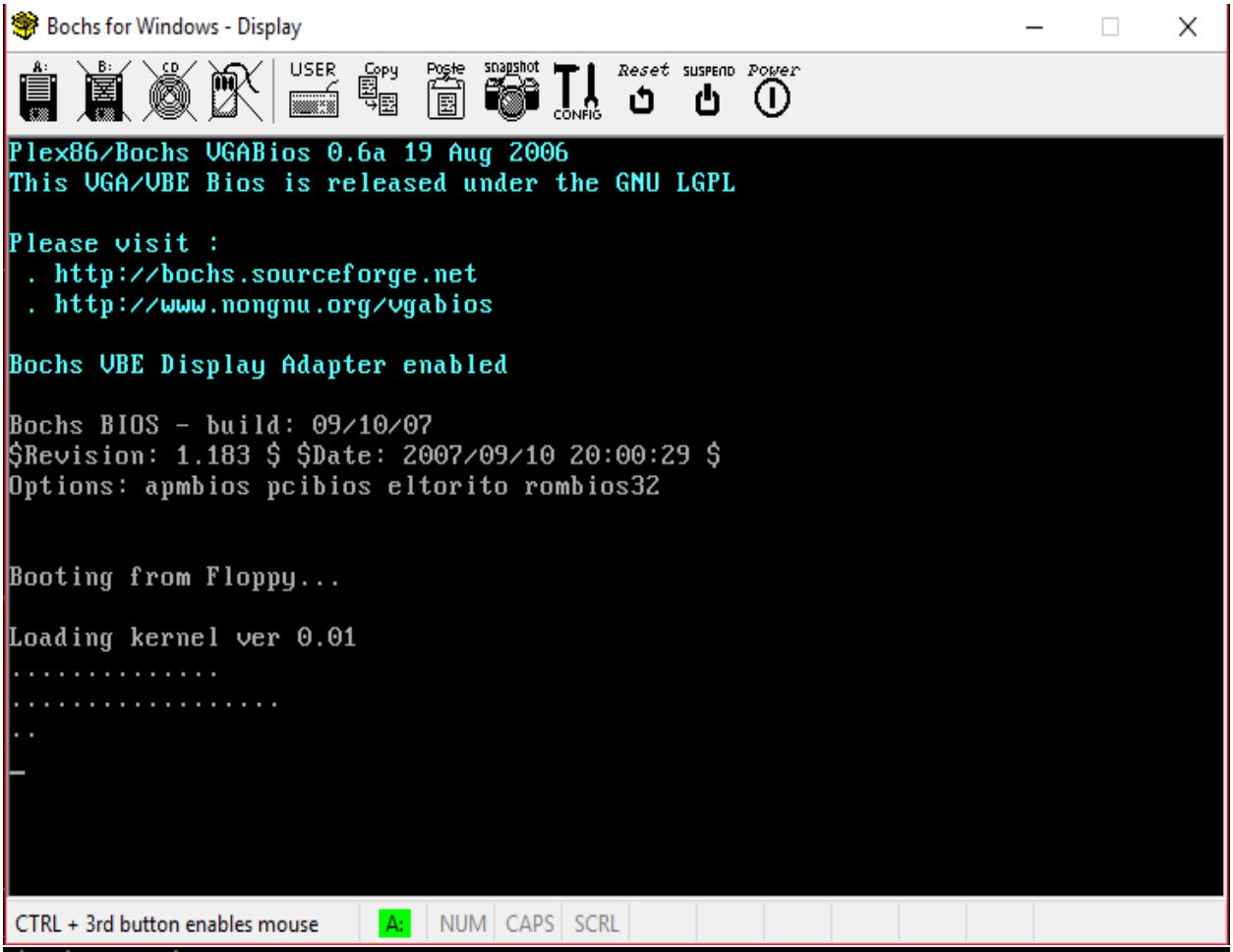
boot.asm - Notepad

File Edit Format View Help

```
    ; Mengatur lokasi kode program pada alamat 7C00:0000, dan mengatur REGISTER S
        cli                                     ; matikan aktifitas interupsi
        mov     ax, 0x07C0
        mov     ds, ax
        mov     es, ax
        mov     fs, ax
        mov     gs, ax

    ; Mengatur lokasi stack
        mov     ax, 0x0000
        mov     ss, ax
        mov     sp, 0xFFFF    ; sp bergerak dari alamat atas ke bawah
        sti                                     ; aktifkan aktifitas interupsi


    ; Menampilkan text di layar
        mov     si,msgLoading         ; mengambil lokasi text yang di simpan dal
        call     DisplayMessage


    LOAD_ROOT:
    ; menghitung ukuran 'root directory' dan menyimpannya dalam register ‘cx’
        xor     cx,cx
        xor     dx,dx
        mov     ax,0x0020                ; Ukuran satu nama direktori sepanjang
        mul     WORD [MaxRootEntries]              ; Total lokkasi direktori 32
        div     WORD [BytesPerSector]              ; lokasi sektor yang digunak
        xchg    ax, cx                                   ; Ukuran 'root direkto

    ; menghitung jumlah sektor yang digunakan untuk menyimpan FAT
```

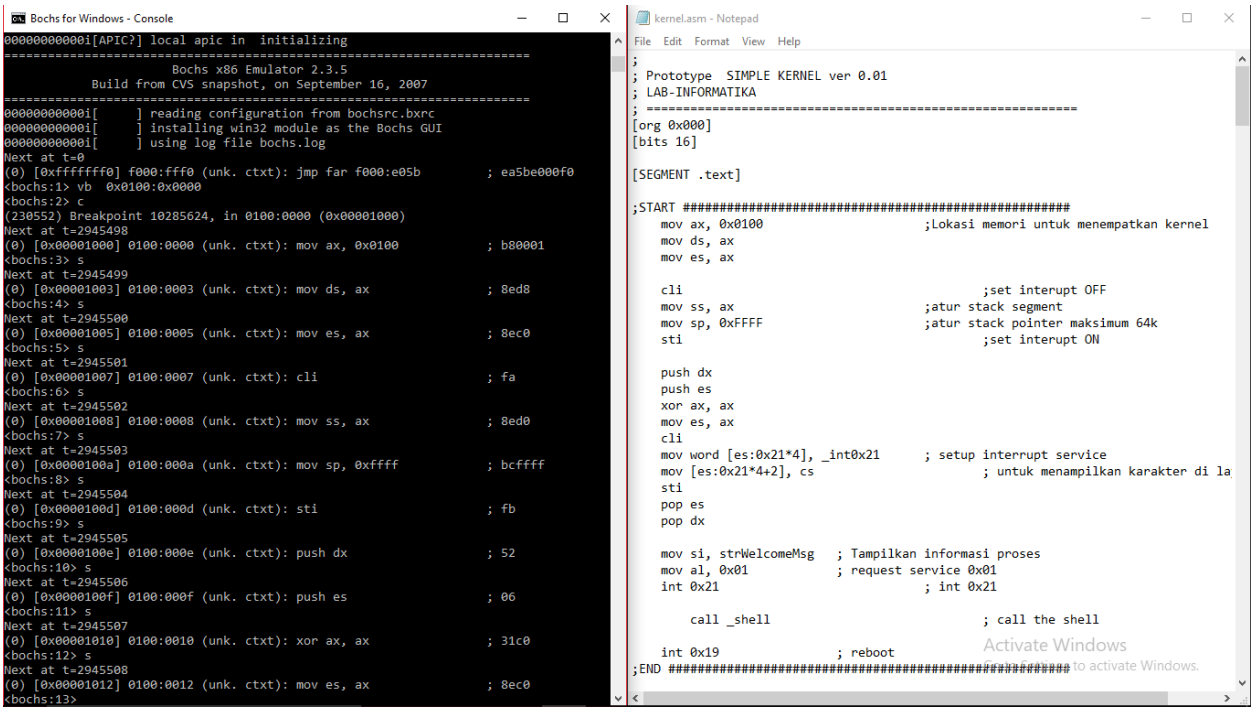
8. Ketikan perintah ‘s’ untuk kembali masuk ke PC simulator

```
C:\OS\LAB\LAB3>s

C:\OS\LAB\LAB3>..\..\bochs-2.3.5\bochsdbg -q -f bochsrc.bxrc
00000000000i[APIC?] local apic in  initializing
========================================================================
                    Bochs x86 Emulator 2.3.5
            Build from CVS snapshot, on September 16, 2007
========================================================================
00000000000i[     ] reading configuration from bochsrc.bxrc
00000000000i[     ] installing win32 module as the Bochs GUI
00000000000i[     ] using log file bochs.log
Next at t=0
(0) [0xfffffff0] f000:fff0 (unk. ctxt): jmp far f000:e05b         ; ea5be000f0
```




```
<bochs:1> vb  0x0100:0x0000
<bochs:2> c
(230552) Breakpoint 10285624, in 0100:0000 (0x00001000)
Next at t=2945498
(0) [0x00001000] 0100:0000 (unk. ctxt): mov ax, 0x0100          ; b80001
<bochs:3> s
```

9. Bandingkan kembali proses debugging dengan file kernel.asm



Bochs for Windows - Console

```
00000000000i[APIC?] local apic in  initializing
========================================================================
                    Bochs x86 Emulator 2.3.5
            Build from CVS snapshot, on September 16, 2007
========================================================================
00000000000i[     ] reading configuration from bochsrc.bxrc
00000000000i[     ] installing win32 module as the Bochs GUI
00000000000i[     ] using log file bochs.log
Next at t=0
(0) [0xfffffff0] f000:fff0 (unk. ctxt): jmp far f000:e05b         ; ea5be000f0
<bochs:1> vb  0x0100:0x0000
<bochs:2> c
(230552) Breakpoint 10285624, in 0100:0000 (0x00001000)
Next at t=2945498
(0) [0x00001000] 0100:0000 (unk. ctxt): mov ax, 0x0100            ; b80001
<bochs:3> s
Next at t=2945499
(0) [0x00001003] 0100:0003 (unk. ctxt): mov ds, ax                ; 8ed8
<bochs:4> s
Next at t=2945500
(0) [0x00001005] 0100:0005 (unk. ctxt): mov es, ax                ; 8ec0
<bochs:5> s
Next at t=2945501
(0) [0x00001007] 0100:0007 (unk. ctxt): cli                       ; fa
<bochs:6> s
Next at t=2945502
(0) [0x00001008] 0100:0008 (unk. ctxt): mov ss, ax                ; 8ed0
<bochs:7> s
Next at t=2945503
(0) [0x0000100a] 0100:000a (unk. ctxt): mov sp, 0xffff            ; bcffff
<bochs:8> s
Next at t=2945504
(0) [0x0000100d] 0100:000d (unk. ctxt): sti                       ; fb
<bochs:9> s
Next at t=2945505
(0) [0x0000100e] 0100:000e (unk. ctxt): push dx                   ; 52
<bochs:10> s
Next at t=2945506
(0) [0x0000100f] 0100:000f (unk. ctxt): push es                   ; 06
<bochs:11> s
Next at t=2945507
(0) [0x00001010] 0100:0010 (unk. ctxt): xor ax, ax                ; 31c0
<bochs:12> s
Next at t=2945508
(0) [0x00001012] 0100:0012 (unk. ctxt): mov es, ax                ; 8ec0
<bochs:13>
```

kernel.asm - Notepad

File Edit Format View Help

```
;
; Prototype  SIMPLE KERNEL ver 0.01
; LAB-INFORMATIKA
; =========================================================
[org 0x000]
[bits 16]

[SEGMENT .text]

;START ##################################################
    mov ax, 0x0100                      ;Lokasi memori untuk menempatkan kernel
    mov ds, ax
    mov es, ax

    cli                                         ;set interupt OFF
    mov ss, ax                          ;atur stack segment
    mov sp, 0xFFFF                      ;atur stack pointer maksimum 64k
    sti                                         ;set interupt ON

    push dx
    push es
    xor ax, ax
    mov es, ax
    cli
    mov word [es:0x21*4], _int0x21      ; setup interrupt service
    mov [es:0x21*4+2], cs                       ; untuk menampilkan karakter di la
    sti
    pop es
    pop dx

    mov si, strWelcomeMsg   ; Tampilkan informasi proses
    mov al, 0x01            ; request service 0x01
    int 0x21                            ; int 0x21

        call _shell                             ; call the shell

    int 0x19                ; reboot
;END ####################################################
```