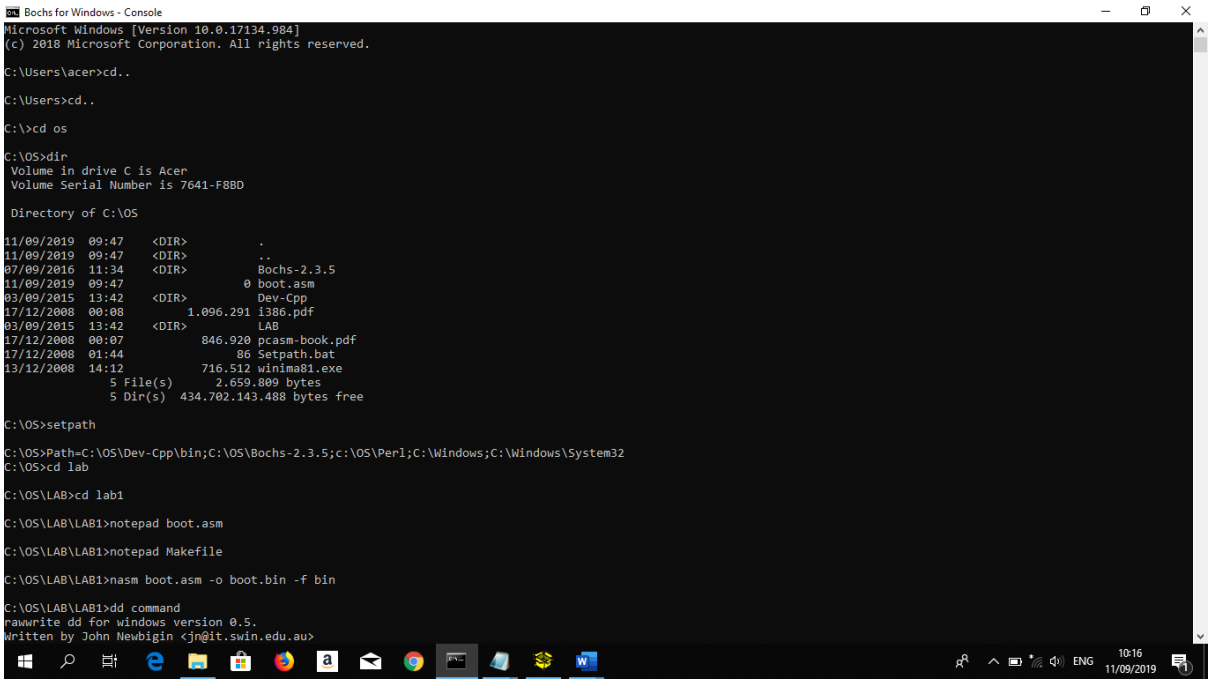
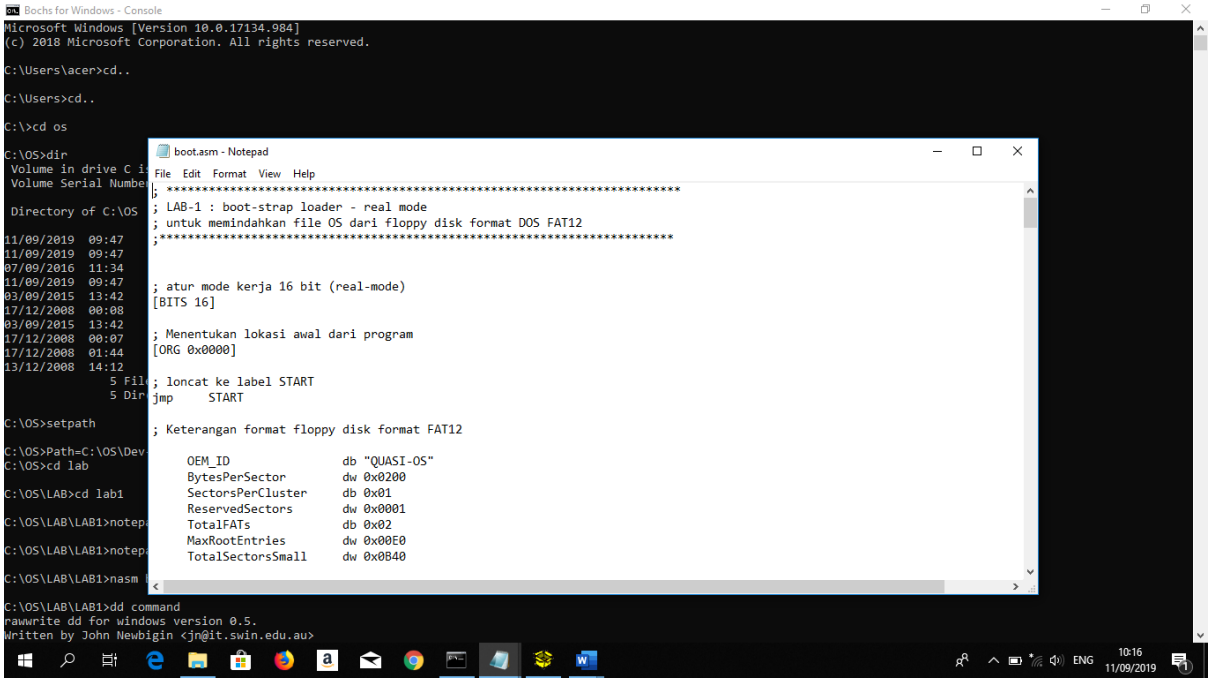
NAMA : Rizzky Prasetyo

NIM : L200180032





boot.asm - Notepad

File Edit Format View Help

```
; ********************************************************************************
; LAB-1 : boot-strap loader - real mode
; untuk memindahkan file OS dari floppy disk format DOS FAT12
;********************************************************************************


; atur mode kerja 16 bit (real-mode)
[BITS 16]

; Menentukan lokasi awal dari program
[ORG 0x0000]

; loncat ke label START
jmp     START

; Keterangan format floppy disk format FAT12

     OEM_ID               db "QUASI-OS"
     BytesPerSector       dw 0x0200
     SectorsPerCluster    db 0x01
     ReservedSectors      dw 0x0001
     TotalFATs            db 0x02
     MaxRootEntries       dw 0x00E0
     TotalSectorsSmall    dw 0x0B40
     MediaDescriptor      db 0xF0
     SectorsPerFAT        dw 0x0009
     SectorsPerTrack      dw 0x0012
     NumHeads             dw 0x0002
     HiddenSectors        dd 0x00000000
     TotalSectorsLarge    dd 0x00000000
     DriveNumber          db 0x00
     Flags                db 0x00
     Signature            db 0x29
     VolumeID             dd 0xFFFFFFFF
     VolumeLabel          db "QUASI  BOOT"
     SystemID             db "FAT12   "
```

boot.asm - Notepad

File Edit Format View Help

```
     START:
     ; Mengatur lokasi kode program pada alamat 7C00:0000, dan mengatur REGISTER SEGMEN
          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 dalam 'msgloading'
          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 32 byte
          mul     WORD [MaxRootEntries]              ; Total lokkasi direktori 32 x 224 (heksa 0x00E0)=7168
          div     WORD [BytesPerSector]              ; lokasi sektor yang digunakan untuk menyimpan direktori 7168/512 (0x0200) = 14
          xchg    ax, cx                                   ; Ukuran 'root direktori' = 14 sektor

     ; menghitung jumlah sektor yang digunakan untuk menyimpan FAT
     ; untuk mencari lokasi awal sektor ROOT DIREKTORI di simpan di register 'ax'
     ; hasil disimpan pada variabel 'datasector'
          mov     al, BYTE [TotalFATs]               ; jumlah FAT (2 copy)
          mul     WORD [SectorsPerFAT]               ; dikalikan dengan jumlah sektor yang digunakan FAT (9 sektor/FAT)
          add     ax, WORD [ReservedSectors]         ; tambah cadangan sektor (1 sektor) simpan di 'ax'
```

boot.asm - Notepad

File Edit Format View Help

```
        add     ax, WORD [ReservedSectors]          ; tambah cadangan sektor (1 sektor) simpan di 'ax'
        mov     WORD [datasector], ax               ; pindahkan hasilnya ke variabel 'datasector'(=19 sektor)
        add     WORD [datasector], cx               ; tambah dengan ukuran 'root direktori' (14),
                                                    ; jadi total 33 sektor untuk menyimpan 2 FAT (Field Allocation Table)
                                                    ; dan data 'ROOT DIREKTORI'

    ;
    ; Memindahkan isi root direktori ke memori RAM pada alamat 7C00:0200
    ;

        mov     bx, 0x0200                          ; lokasi awal memori untuk menampung pembacaan sektor
        call    ReadSectors

    ;
    ; mencari nama file OS
    ;
        mov     cx, WORD [MaxRootEntries]           ; baca semua file dalam root direktori
        mov     di, 0x0200                          ; lokasi awal berisi copy ROOT DIREKTORI

    .LOOP:
        push    cx
        mov     cx, 0x000B                          ; ukuran nama FILE dan DIREKTORI sebanyak 11 karakter
        mov     si, ImageName                       ; Mengambil nama FILE yang di cari
                                                    ; nama FILE tersimpan dalam variabel ImageName
                                                    ; file diberi nama 'MYKERNELBIN'
        push    di
    rep cmpsb                                       ; Bandingkan dengan daftar file root direktori
        pop     di
        je      LOAD_FAT                            ; jika file sudah ditemukan loncat ke label LOAD_FAT
        pop     cx
        add     di, 0x0020                          ; Nama file berikutnya
        loop    .LOOP                               ; periksa lagi
        jmp     FAILURE                             ; JIka nama file tidak ada loncat ke label FAILURE


    LOAD_FAT:
    ;
```

boot.asm - Notepad

File Edit Format View Help

```
    LOAD_FAT:
    ;
    ; Simpan nomor awal cluster tempat menyimpan data file 'MYKERNELBIN'
        mov     si, msgCRLF
        call    DisplayMessage
        mov     dx, WORD [di + 0x001A]
        mov     WORD [cluster], dx                  ; Cluster pertama file 'MYKERNELBIN',
                                                    ; byte ke 0x1A (26) dari 32 byte, dibelakang nama file
                                                    ; register 'di' menunjuk lokasi byte pertama dari namafile
    ; Menghitung jumlah sektor yang digunakan untuk menyimpan FAT
    ; hasilnya disimpan pada register 'cx'
        xor     ax, ax
        mov     al, BYTE [TotalFATs]                ; Jumlah FAT, pada FAT12 ada 2 copy FAT
        mul     WORD [SectorsPerFAT]                ; dikalikan dengan jumlah sektor/FAT (9)
        mov     cx, ax

    ; Mencari nomor awal sektor lokasi FAT
    ; hasil disimpan di reg 'ax'
        mov     ax, WORD [ReservedSectors]          ; sesuaikan dengan jumlah sektor cadangan (1 sektor)

    ; baca isi FAT pindahkan ke lokasi memory (7C00:0200)
        mov     bx, 0x0200                          ; pindahkan FAT di atas lokasi bootcode
        call    ReadSectors

    ; baca isi file 'KERNEL  BIN' pindahkan ke memori dengan lokasi (0100:0000)
        mov     si, msgCRLF
        call    DisplayMessage                      ; menampilkan text di layar
        mov     ax, 0x0100                          ; atur segmen kode target CS = 0x0100
        mov     es, ax
        mov     bx, 0x0000                          ; atur target IP= 0x0000
        push    bx

    LOAD_IMAGE:
        mov     ax, WORD [cluster]                  ; Nomor cluster awal 'KERNEL  BIN'
        pop     bx                                  ; Lokasi awal memori untuk menyimpan 'KERNEL  BIN'
        call    ClusterLBA                          ; konversi dagta cluster to LBA (nomor sektor)
                                                    ; hasil di simpan di 'ax' (nomor sektor yang harus di baca)
```

boot.asm - Notepad

File Edit Format View Help

```
                                        ; hasil di simpan di 'ax' (nomor sektor yang harus di baca)
        xor     cx, cx
        mov     cl, BYTE [SectorsPerCluster]    ; Jumlah sektor yang di baca
        call    ReadSectors
        push    bx

; Baca cluster berikutnya
        mov     ax, WORD [cluster]              ; identifikasi posisi cluster saat ini
        mov     cx, ax                          ; copy cluster saat ini ke 'cx'
        mov     dx, ax                          ; copy cluster saat ini ke 'dx'
        shr     dx, 0x0001                      ; bagi dengan 2 (posisi bit di geser ke kanan 1 digit)

; nomor sekarang di tambah (1/2 nomor cluster)
        add     cx, dx                          ; (3/2) nilai cluster saat ini
        mov     bx, 0x0200                      ; lokasi FAT di memory
        add     bx, cx                          ; index FAT
        mov     dx, WORD [bx]                   ; baca dua byte dari FAT
        test    ax, 0x0001
        jnz     .ODD_CLUSTER
.EVEN_CLUSTER:
        and     dx, 0000111111111111b           ; jika angkanya genap ambil 12 bit terbawah
       jmp     .DONE
.ODD_CLUSTER:
        shr     dx, 0x0004                      ; jika angkanya ganjil ambil 12 bit teratas
.DONE:
        mov     WORD [cluster], dx              ; simpan nomor cluster yang baru
        cmp     dx, 0x0FF0                      ; Periksa sudah mencapai bagian akhir file
        jb      LOAD_IMAGE
DONE:
        mov     si, msgCRLF
        call    DisplayMessage
        push    WORD 0x0100                     ; Loncat ke alamat 0100:0000
        push    WORD 0x0000
        retf
FAILURE:
        mov     si, msgFailure
        call    DisplayMessage
```

boot.asm - Notepad

File Edit Format View Help

```
FAILURE:
        mov     si, msgFailure
        call    DisplayMessage
        mov     ah, 0x00
        int     0x16                            ; Tunggu penekanan tombol keyboard
        int     0x19                            ; reboot

;*************************************************************************
; PROCEDURE DisplayMessage
; Menampilkan string yang tersimpan pada alamat yang ditunjuk oleh  'ds:si'
; menggunakan program pada BIOS
;*************************************************************************
DisplayMessage:
        lodsb                                   ; Ambil karakter berikutnya
        or      al, al                          ; periksa bagian akhir data (apakah karakter NUL?)
        jz      .DONE
        mov     ah, 0x0E                        ; BIOS teletype
        mov     bh, 0x00                        ; display page 0
        mov     bl, 0x07                        ; text attribute
        int     0x10                            ; panggil BIOS
        jmp     DisplayMessage
.DONE:
        ret

;*************************************************************************
; PROSEDURE ReadSectors
; Membaca sektor sebanyak angka yang tersimpan pada register ‘cx’
; mulai dari nomor sektor, sesuai angka yang tersimpan pada register 'ax'
; dipindahkan ke lokasi memori yg ditunjuk oleh kombinasi register ‘es:bx’
;*************************************************************************
ReadSectors:                                    ; nama prosedure
.MAIN
        mov     di, 0x0005                      ; mencoba membaca sektor sebanyak 5 kali
                                                ; untuk memastikan adanya Error pada DISK
.SECTORLOOP
        push    ax
        push    bx
```

boot.asm - Notepad

File Edit Format View Help

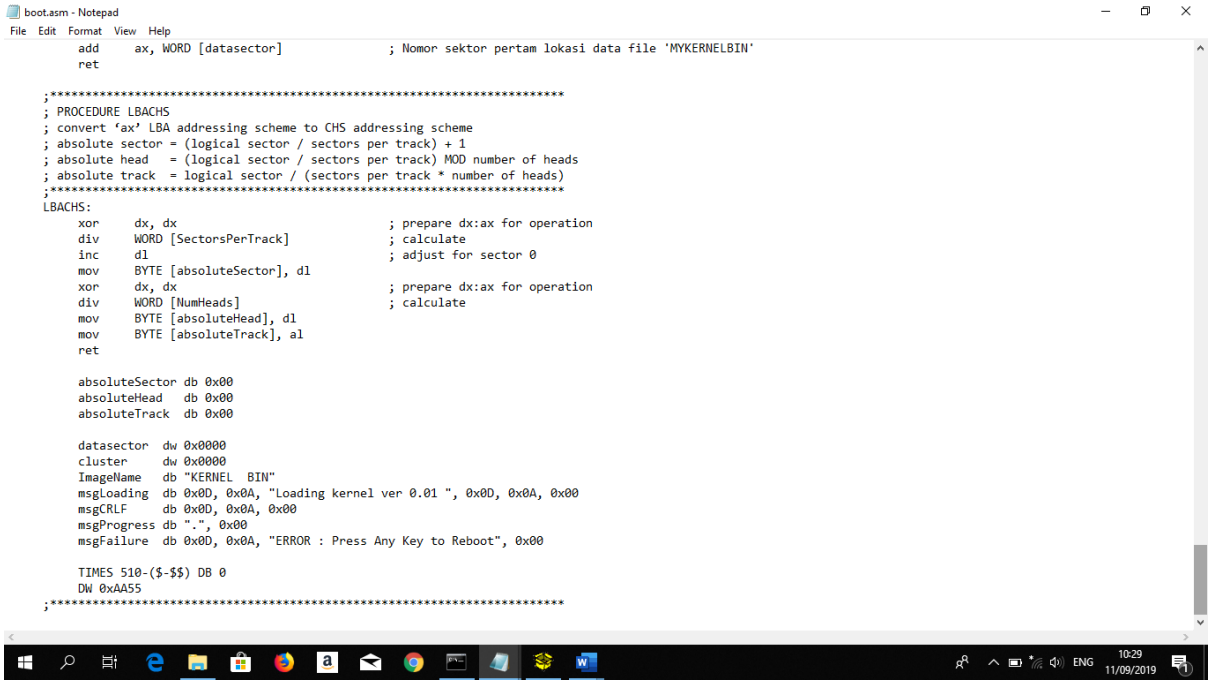
```
    .SECTORLOOP
        push    ax
        push    bx
        push    cx
        call    LBACHS
        mov     ah, 0x02                        ; Perintah BIOS untuk membaca sektor
        mov     al, 0x01                        ; jumlah sektor yang di baca (1 sektor)
        mov     ch, BYTE [absoluteTrack]        ; Nomor track, absoluteTrack=0
        mov     cl, BYTE [absoluteSector]       ; Nomor sektor, absoluteSector = 0
        mov     dh, BYTE [absoluteHead]         ; Nomor head, absoluteHead = 0, head ada 2 dengan nomor 0 dan 1
        mov     dl, BYTE [DriveNumber]          ; Nomor drive, 'DriveNumber=0
        int     0x13                            ; memanggil BIOS untuk membaca sektor

        jnc     .SUCCESS                        ; Tes hasil pembacaan sektor
                                                ; jika C=0 (sukses) loncat ke label '.SUCCESS'
        xor     ax, ax                          ; pilih fungsi BIOS untuk mereset disk
        int     0x13                            ; panggil BIOS
        dec     di                              ; kurangi angka jumlah pengujian baca sektor
        pop     cx
        pop     bx
        pop     ax
        jnz     .SECTORLOOP                     ; jika ada error sebelumnya coba baca lagi
        int     0x18
    .SUCCESS
        mov     si, msgProgress                 ; menampilkan status proses baca sektor
        call    DisplayMessage
        pop     cx
        pop     bx
        pop     ax
        add     bx, WORD [BytesPerSector]       ; lokasi memori selanjutnya
        inc     ax                              ; nomor sektor berikutnya
        loop    .MAIN                           ; baca sector berikutnya
        ret

    ;*************************************************************************
    ; PROSEDURE ClusterLBA
    ; Merubah data nomor cluster menjadi skema alamat LBA
```

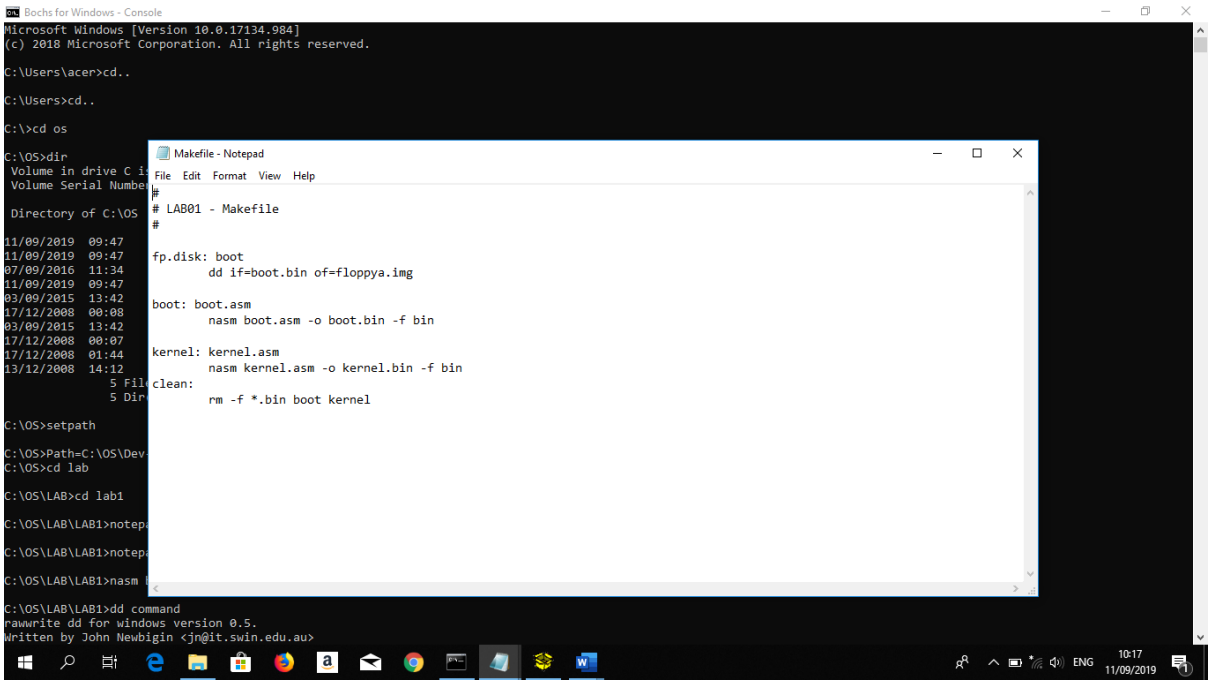
boot.asm - Notepad

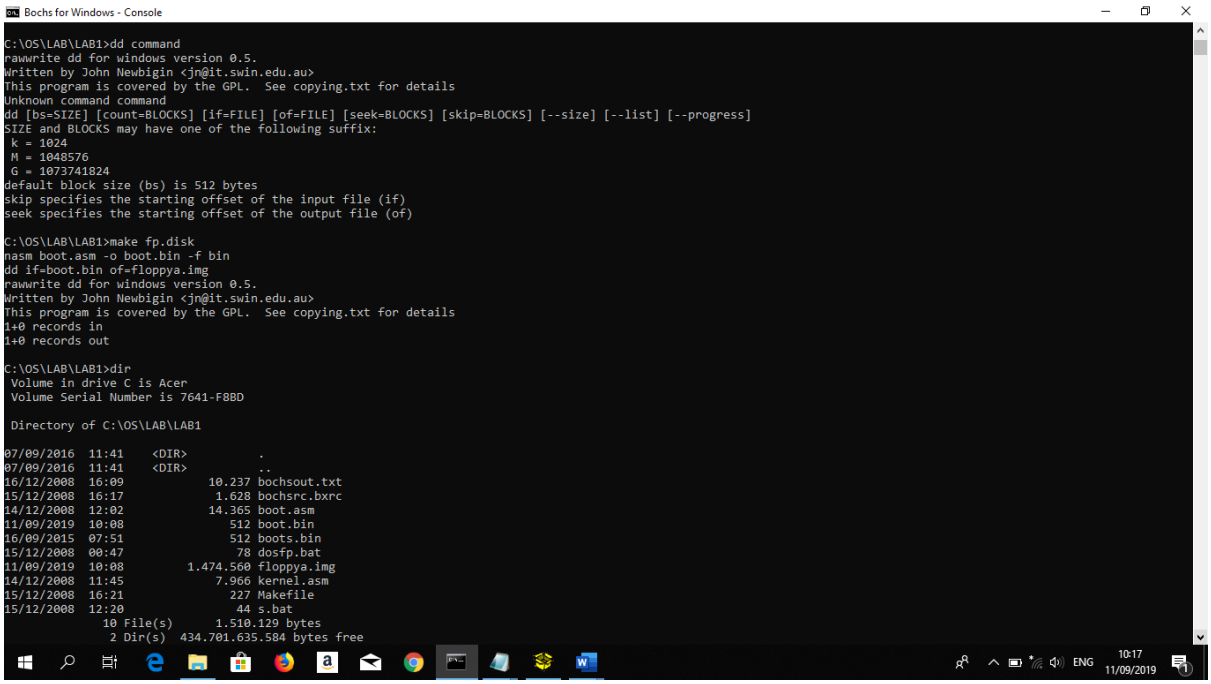
File Edit Format View Help

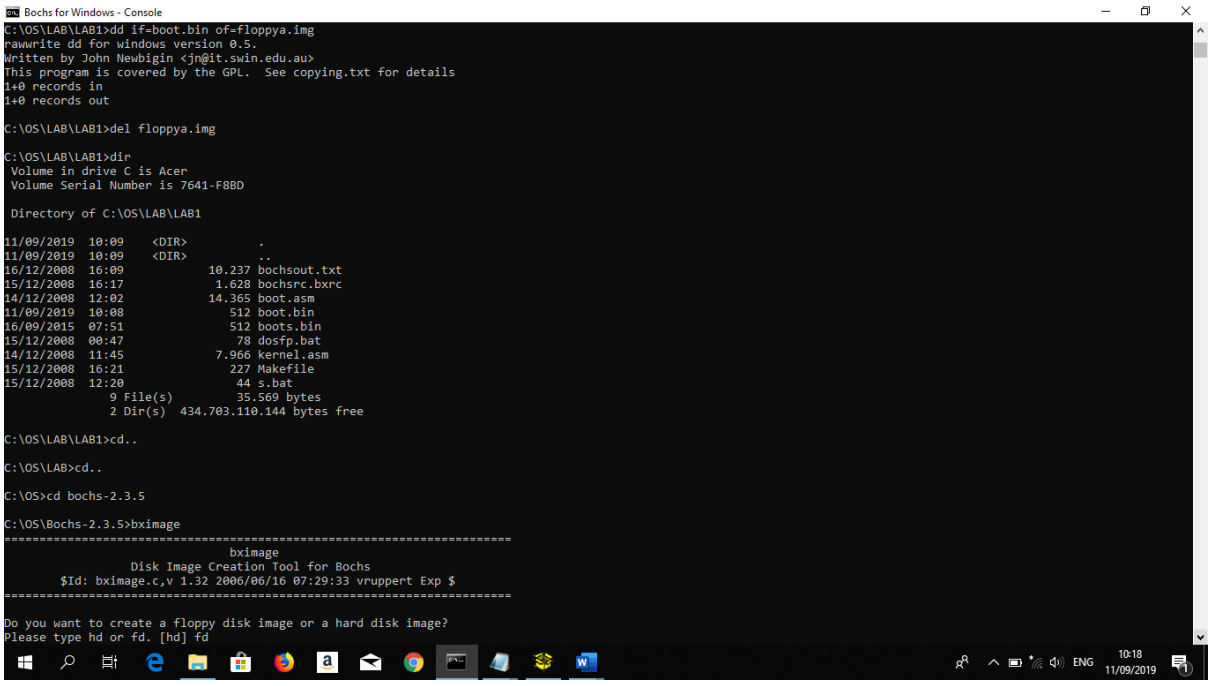
```
    ;*************************************************************************
    ; PROSEDURE ClusterLBA
    ; Merubah data nomor cluster menjadi skema alamat LBA
    ; dengan rumus sebagai berikut
    ; LBA = (cluster - 2) * sectors per cluster
    ; nomor cluster di simpan pada register 'ax'
    ;*************************************************************************
    ClusterLBA:
        sub     ax, 0x0002                      ; nomor cluster - 2, hasil disimpan di 'ax'
        xor     cx, cx
        mov     cl, BYTE [SectorsPerCluster]    ; konversi dari byte to word
        mul     cx
        add     ax, WORD [datasector]           ; Nomor sektor pertam lokasi data file 'MYKERNELBIN'
        ret

    ;*************************************************************************
    ; PROCEDURE LBACHS
    ; convert ‘ax’ LBA addressing scheme to CHS addressing scheme
    ; absolute sector = (logical sector / sectors per track) + 1
    ; absolute head   = (logical sector / sectors per track) MOD number of heads
    ; absolute track  = logical sector / (sectors per track * number of heads)
    ;*************************************************************************
    LBACHS:
        xor     dx, dx                          ; prepare dx:ax for operation
        div     WORD [SectorsPerTrack]          ; calculate
        inc     dl                              ; adjust for sector 0
        mov     BYTE [absoluteSector], dl
        xor     dx, dx                          ; prepare dx:ax for operation
        div     WORD [NumHeads]                 ; calculate
        mov     BYTE [absoluteHead], dl
        mov     BYTE [absoluteTrack], al
        ret

        absoluteSector db 0x00
        absoluteHead   db 0x00
        absoluteTrack  db 0x00
```

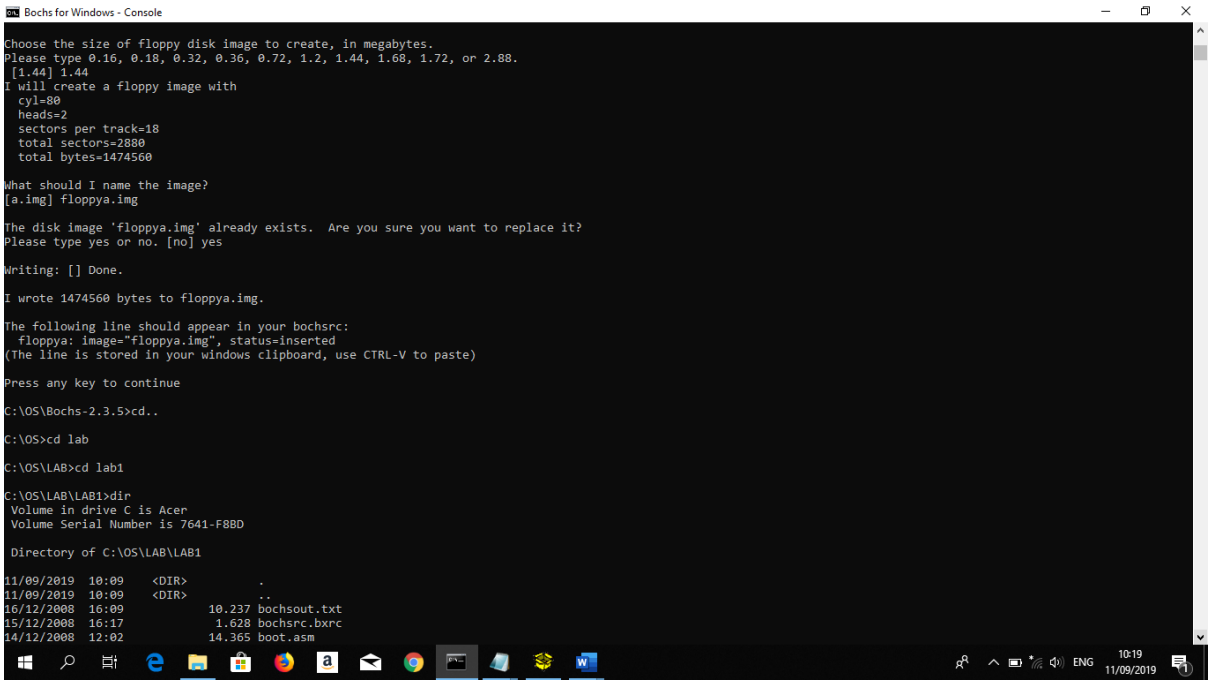
ENG 10:28 11/09/2019

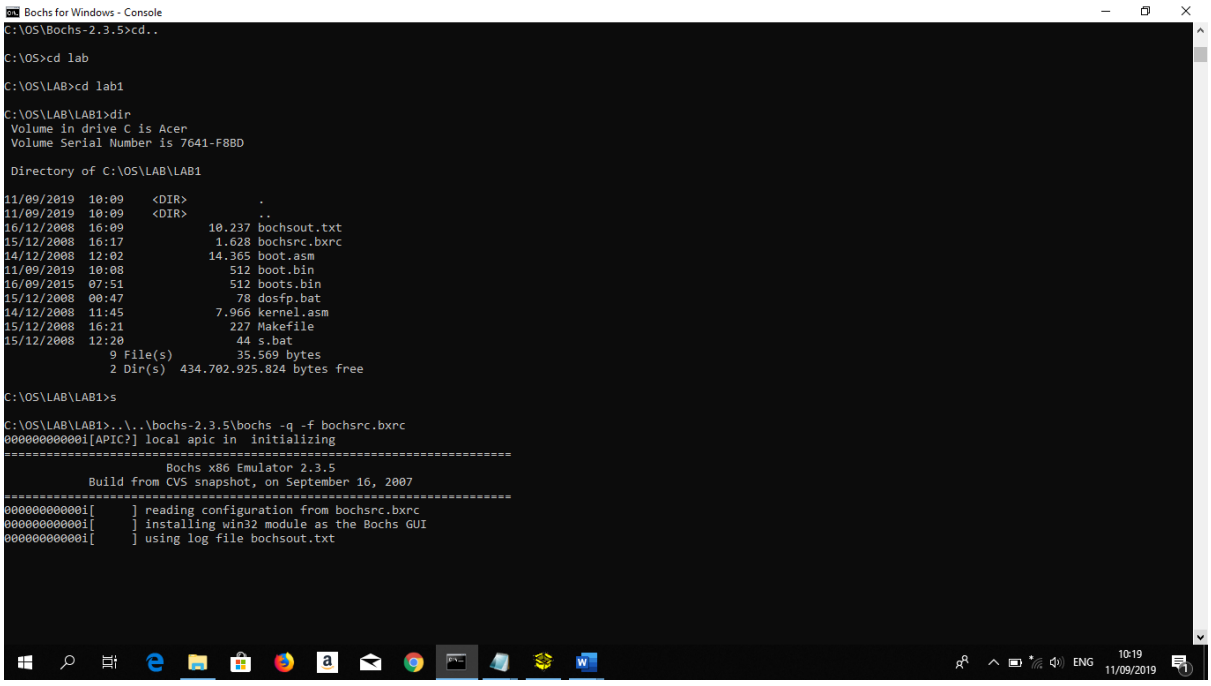

boot.asm - Notepad
File Edit Format View Help
add ax, WORD [datasector] ; Nomor sektor pertam lokasi data file 'MYKERNELBIN'
ret
;***
; PROCEDURE LBACHS
; convert 'ax' LBA addressing scheme to CHS addressing scheme
; absolute sector = (logical sector / sectors per track) + 1
; absolute head = (logical sector / sectors per track) MOD number of heads
; absolute track = logical sector / (sectors per track * number of heads)
;***
LBACHS:
xor dx, dx ; prepare dx:ax for operation
div WORD [SectorsPerTrack] ; calculate
inc dl ; adjust for sector 0
mov BYTE [absoluteSector], dl
xor dx, dx ; prepare dx:ax for operation
div WORD [NumHeads] ; calculate
mov BYTE [absoluteHead], dl
mov BYTE [absoluteTrack], al
ret
absoluteSector db 0x00
absoluteHead db 0x00
absoluteTrack db 0x00
datasector dw 0x0000
cluster dw 0x0000
ImageName db "KERNEL BIN"
msgLoading db 0x0D, 0x0A, "Loading kernel ver 0.01 ", 0x0D, 0x0A, 0x00
msgCRLF db 0x0D, 0x0A, 0x00
msgProgress db ".", 0x00
msgFailure db 0x0D, 0x0A, "ERROR : Press Any Key to Reboot", 0x00
TIMES 510-($-$$) DB 0
DW 0xAA55
;***
ENG 10:29 11/09/2019

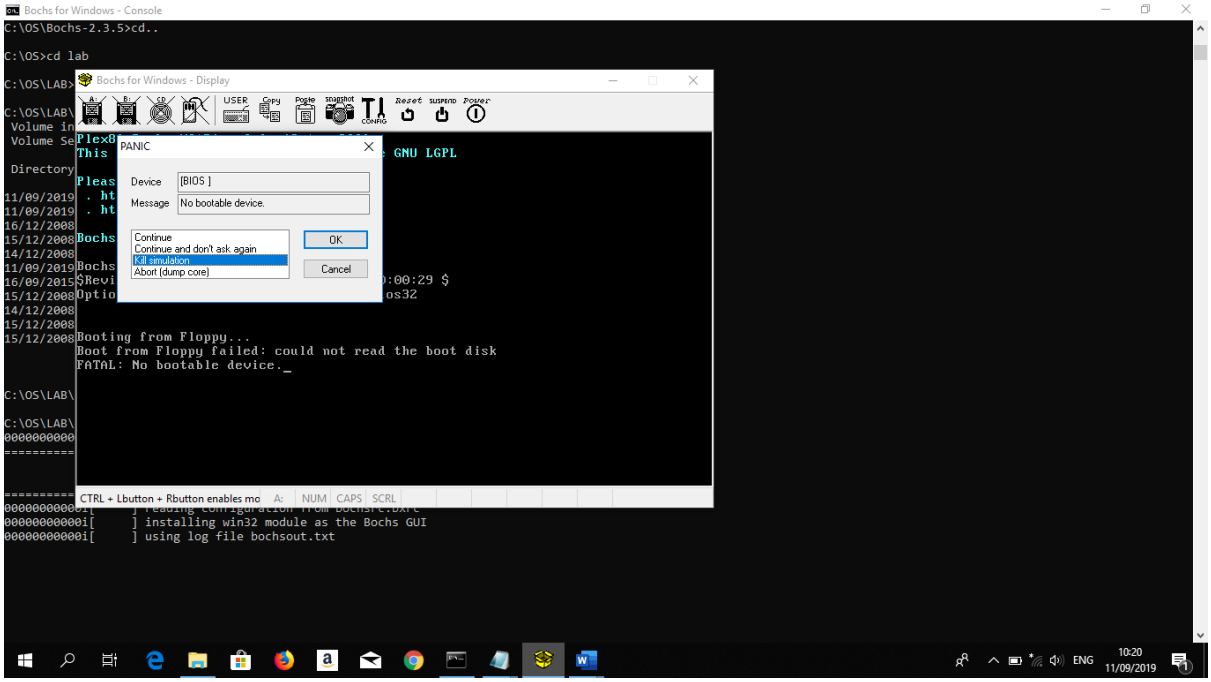

Bochs for Windows - Console
Microsoft Windows [Version 10.0.17134.984]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\acer>cd..
C:\Users>cd..
C:\>cd os
C:\OS>dir
Volume in drive C is
Volume Serial Number
Directory of C:\OS
11/09/2019 09:47
11/09/2019 09:47
07/09/2016 11:34
11/09/2019 09:47
03/09/2015 13:42
17/12/2008 00:08
03/09/2015 13:42
17/12/2008 00:07
17/12/2008 01:44
13/12/2008 14:12
5 File
5 Dir
C:\OS>setpath
C:\OS>Path=C:\OS\Dev
C:\OS>cd lab
C:\OS\LAB>cd lab1
C:\OS\LAB\LAB1>notep
C:\OS\LAB\LAB1>notep
C:\OS\LAB\LAB1>nasm
C:\OS\LAB\LAB1>dd command
rawwrite dd for windows version 0.5.
Written by John Newbigin <jn@it.swin.edu.au>
Makefile - Notepad
File Edit Format View Help
#
LAB01 - Makefile
#
fp.disk: boot
dd if=boot.bin of=floppya.img
boot: boot.asm
nasm boot.asm -o boot.bin -f bin
kernel: kernel.asm
nasm kernel.asm -o kernel.bin -f bin
clean:
rm -f *.bin boot kernel
ENG 10:17 11/09/2019


Bochs for Windows - Console
C:\OS\LAB\LAB1>dd command
rawwrite dd for windows version 0.5.
Written by John Newbigin <jn@it.swin.edu.au>
This program is covered by the GPL. See copying.txt for details
Unknown command command
dd [bs=SIZE] [count=BLOCKS] [if=FILE] [of=FILE] [seek=BLOCKS] [skip=BLOCKS] [--size] [--list] [--progress]
SIZE and BLOCKS may have one of the following suffix:
 k = 1024
 M = 1048576
 G = 1073741824
default block size (bs) is 512 bytes
skip specifies the starting offset of the input file (if)
seek specifies the starting offset of the output file (of)
C:\OS\LAB\LAB1>make fp.disk
nasm boot.asm -o boot.bin -f bin
dd if=boot.bin of=floppya.img
rawwrite dd for windows version 0.5.
Written by John Newbigin <jn@it.swin.edu.au>
This program is covered by the GPL. See copying.txt for details
1+0 records in
1+0 records out
C:\OS\LAB\LAB1>dir
 Volume in drive C is Acer
 Volume Serial Number is 7641-F8BD
 Directory of C:\OS\LAB\LAB1
07/09/2016 11:41   <DIR>   .
07/09/2016 11:41   <DIR>   ..
16/12/2008 16:09   10.237 bochsout.txt
15/12/2008 16:17   1.628 bochsrc.bxrc
14/12/2008 12:02   14.365 boot.asm
11/09/2019 10:08   512 boot.bin
16/09/2015 07:51   512 boots.bin
15/12/2008 00:47   78 dosfp.bat
11/09/2019 10:08   1.474.560 floppya.img
14/12/2008 11:45   7.966 kernel.asm
15/12/2008 16:21   227 Makefile
15/12/2008 12:20   44 s.bat
              10 File(s)   1.510.129 bytes
               2 Dir(s) 434.701.635.584 bytes free
ENG
10:17
11/09/2019


Bochs for Windows - Console
C:\OS\LAB\LAB1>dd if=boot.bin of=floppya.img
rawwrite dd for windows version 0.5.
Written by John Newbigin <jn@it.swin.edu.au>
This program is covered by the GPL. See copying.txt for details
1+0 records in
1+0 records out
C:\OS\LAB\LAB1>del floppya.img
C:\OS\LAB\LAB1>dir
 Volume in drive C is Acer
 Volume Serial Number is 7641-F8BD
 Directory of C:\OS\LAB\LAB1
11/09/2019 10:09   <DIR>   .
11/09/2019 10:09   <DIR>   ..
16/12/2008 16:09   10.237 bochsout.txt
15/12/2008 16:17   1.628 bochsrc.bxrc
14/12/2008 12:02   14.365 boot.asm
11/09/2019 10:08   512 boot.bin
16/09/2015 07:51   512 boots.bin
15/12/2008 00:47   78 dosfp.bat
14/12/2008 11:45   7.966 kernel.asm
15/12/2008 16:21   227 Makefile
15/12/2008 12:20   44 s.bat
               9 File(s)   35.569 bytes
               2 Dir(s) 434.703.110.144 bytes free
C:\OS\LAB\LAB1>cd..
C:\OS\LAB>cd..
C:\OS>cd bochs-2.3.5
C:\OS\Bochs-2.3.5>bximage
bximage
Disk Image Creation Tool for Bochs
$Id: bximage.c,v 1.32 2006/06/16 07:29:33 vruppert Exp $
Do you want to create a floppy disk image or a hard disk image?
Please type hd or fd. [hd] fd
ENG
10:18
11/09/2019


Bochs for Windows - Console
Choose the size of floppy disk image to create, in megabytes.
Please type 0.16, 0.18, 0.32, 0.36, 0.72, 1.2, 1.44, 1.68, 1.72, or 2.88.
 [1.44] 1.44
I will create a floppy image with
  cyl=80
  heads=2
  sectors per track=18
  total sectors=2880
  total bytes=1474560
What should I name the image?
[a.img] floppya.img
The disk image 'floppya.img' already exists. Are you sure you want to replace it?
Please type yes or no. [no] yes
Writing: [] Done.
I wrote 1474560 bytes to floppya.img.
The following line should appear in your bochsrc:
  floppya: image="floppya.img", status=inserted
(The line is stored in your windows clipboard, use CTRL-V to paste)
Press any key to continue
C:\OS\Bochs-2.3.5>cd..
C:\OS>cd lab
C:\OS\LAB>cd lab1
C:\OS\LAB\LAB1>dir
 Volume in drive C is Acer
 Volume Serial Number is 7641-F8BD
 Directory of C:\OS\LAB\LAB1
11/09/2019 10:09   <DIR>   .
11/09/2019 10:09   <DIR>   ..
16/12/2008 16:09   10.237 bochsout.txt
15/12/2008 16:17   1.628 bochsrc.bxrc
14/12/2008 12:02   14.365 boot.asm
ENG
10:19
11/09/2019


Bochs for Windows - Console
C:\OS\Bochs-2.3.5>cd..
C:\OS>cd lab
C:\OS\LAB>cd lab1
C:\OS\LAB\LAB1>dir
 Volume in drive C is Acer
 Volume Serial Number is 7641-F8BD
 Directory of C:\OS\LAB\LAB1
11/09/2019 10:09   <DIR>   .
11/09/2019 10:09   <DIR>   ..
16/12/2008 16:09   10.237 bochsout.txt
15/12/2008 16:17   1.628 bochsrc.bxrc
14/12/2008 12:02   14.365 boot.asm
11/09/2019 10:08   512 boot.bin
16/09/2015 07:51   512 boots.bin
15/12/2008 00:47   78 dosfp.bat
14/12/2008 11:45   7.966 kernel.asm
15/12/2008 16:21   227 Makefile
15/12/2008 12:20   44 s.bat
               9 File(s)   35.569 bytes
               2 Dir(s) 434.702.925.824 bytes free
C:\OS\LAB\LAB1>s
C:\OS\LAB\LAB1>..\..\bochs-2.3.5\bochs -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 bochsout.txt
ENG
10:19
11/09/2019


Bochs for Windows - Console
C:\OS\Bochs-2.3.5>cd..
C:\OS>cd lab
Bochs for Windows - Display
PANIC
Device
[BIOS]
Message
No bootable device.
Continue
Continue and don't ask again
Kill simulation
Abort (dump core)
OK
Cancel
Booting from Floppy...
Boot from Floppy failed: could not read the boot disk
FATAL: No bootable device.
CTRL + Lbutton + Rbutton enables mo
NUM
CAPS
SCRL
installing win32 module as the Bochs GUI
using log file bochsout.txt
ENG
10:20
11/09/2019