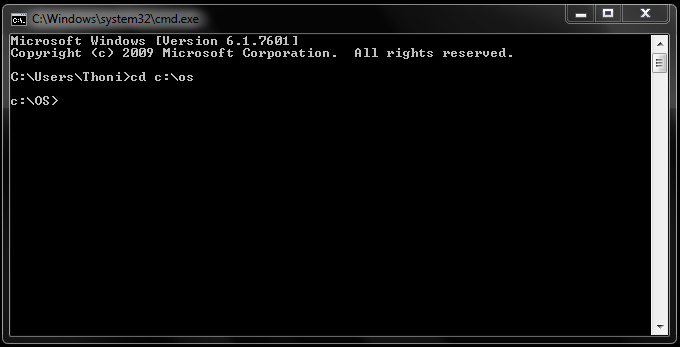
Nama : Kori Indah Prastika Wati

NIM : L200170053

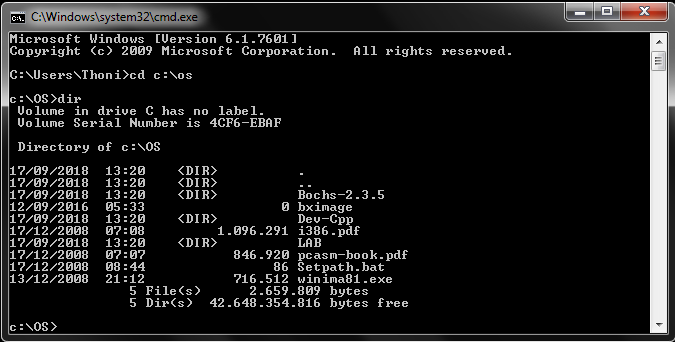
Kelas : C

Laporan Praktikum BAB 1

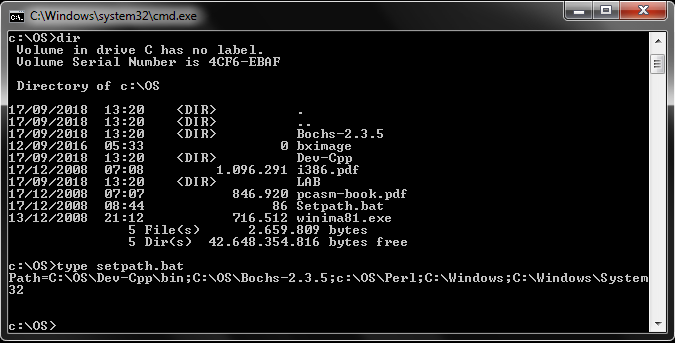
1. Menuju direktori OS



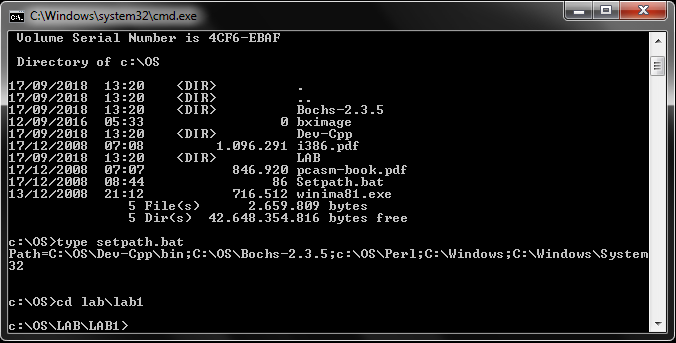
1. Melihat isi direktori OS, dengan mengetikan **dir**



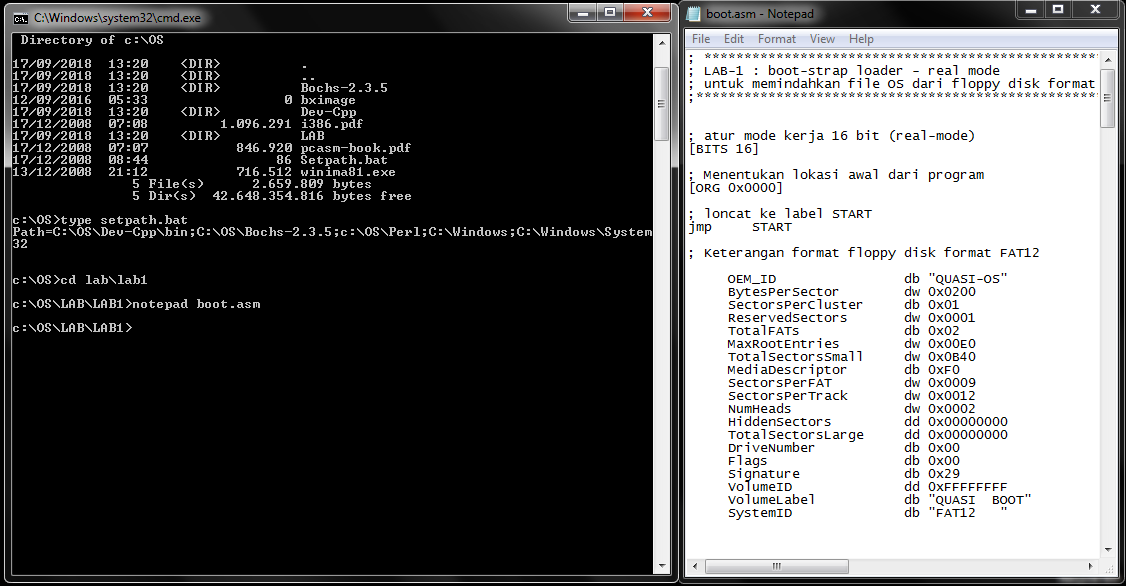
1. Untuk melihat script yang terdapat di dalam file “setpath.bat” dengan mengetikan **type setpath.bat**



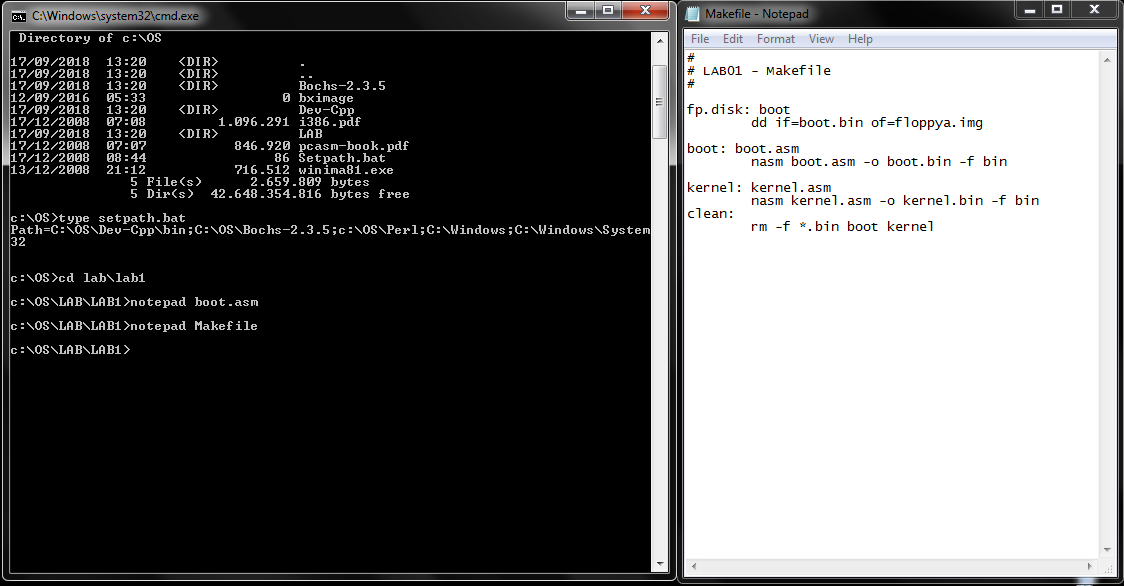
1. Menuju direktori LAB1



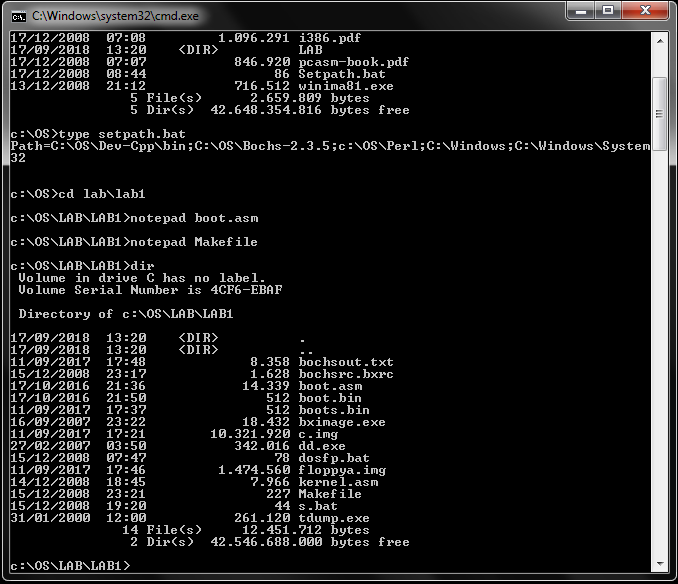
1. Melihat isi boot.asm di notepad, dengan mengetikan **notepad boot.asm**



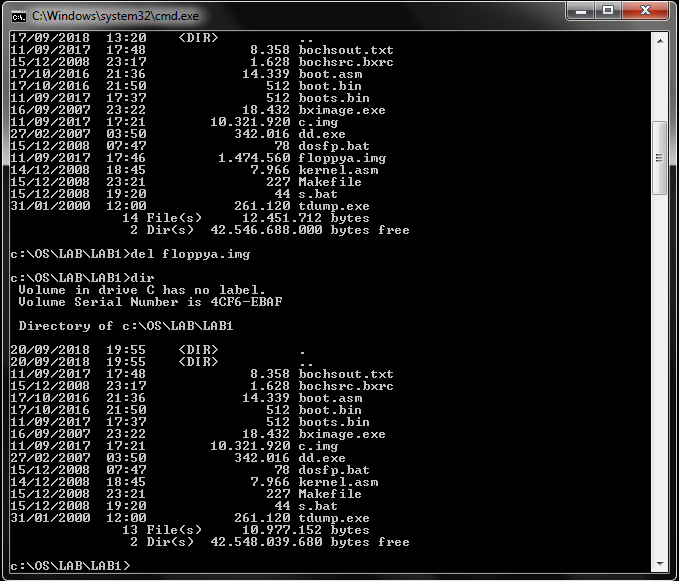
1. Membuka file “Makefile” dengan mengetikkan **Notepad Makefile**



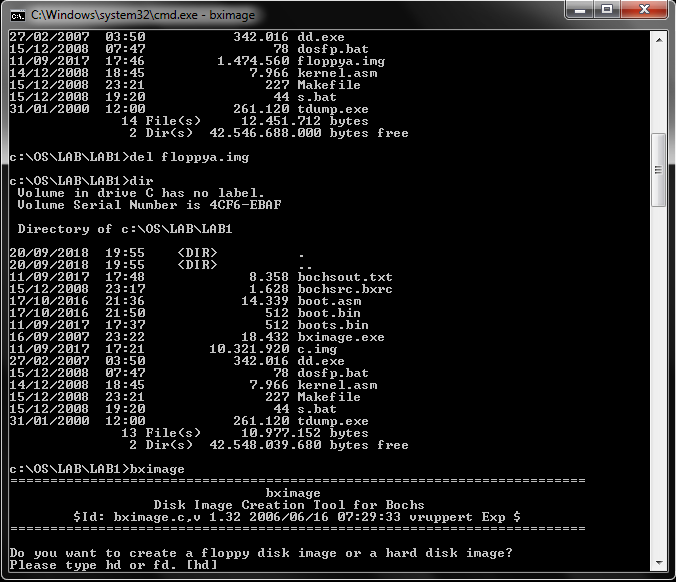
1. Mengecek isi direktori LAB1, dan menghapus boot disk bernama floppya.img



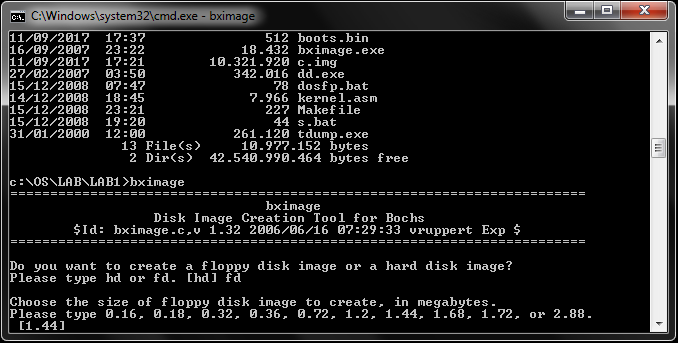
Floppya.img



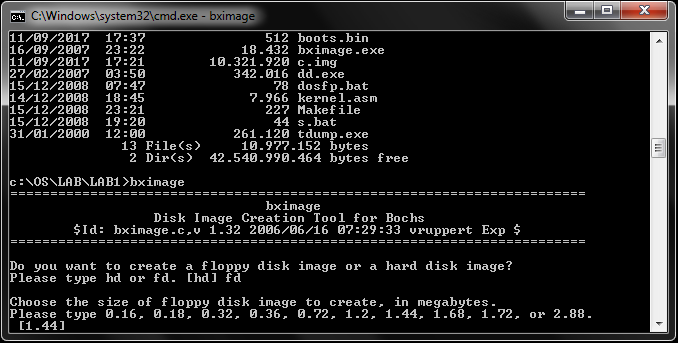
1. Membuat suatu bootdisk baru dengan menggunakan aplikasi bximage.exe. dengan mengetikkan **bximage**



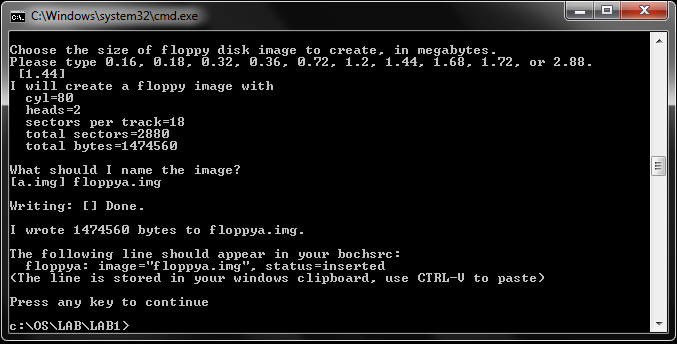
Pilih fd



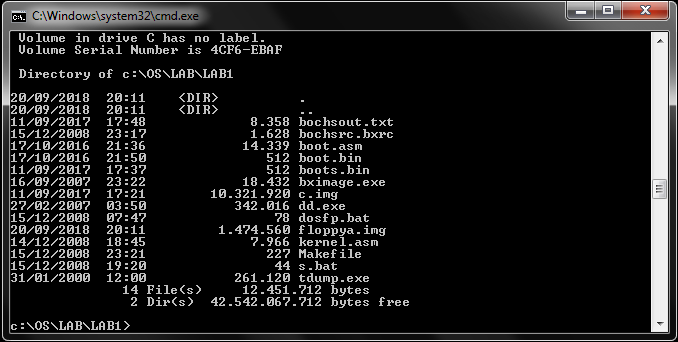
kemudian tekan enter, untuk memilih defaultnya yaitu **[1.44]**



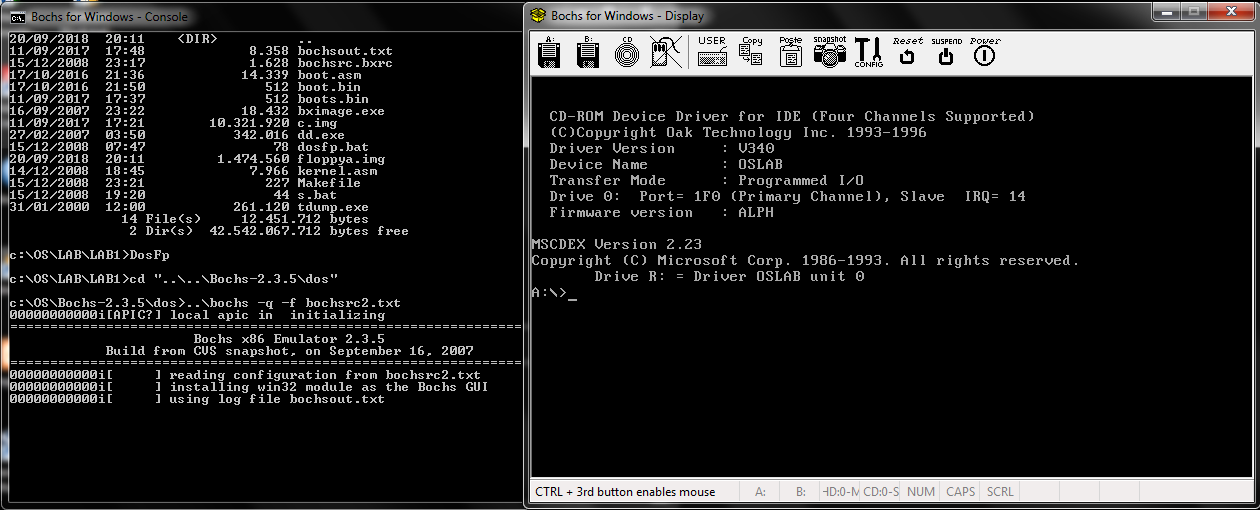
Beri nama floppya.img kemudian tekan enter hingga selesai



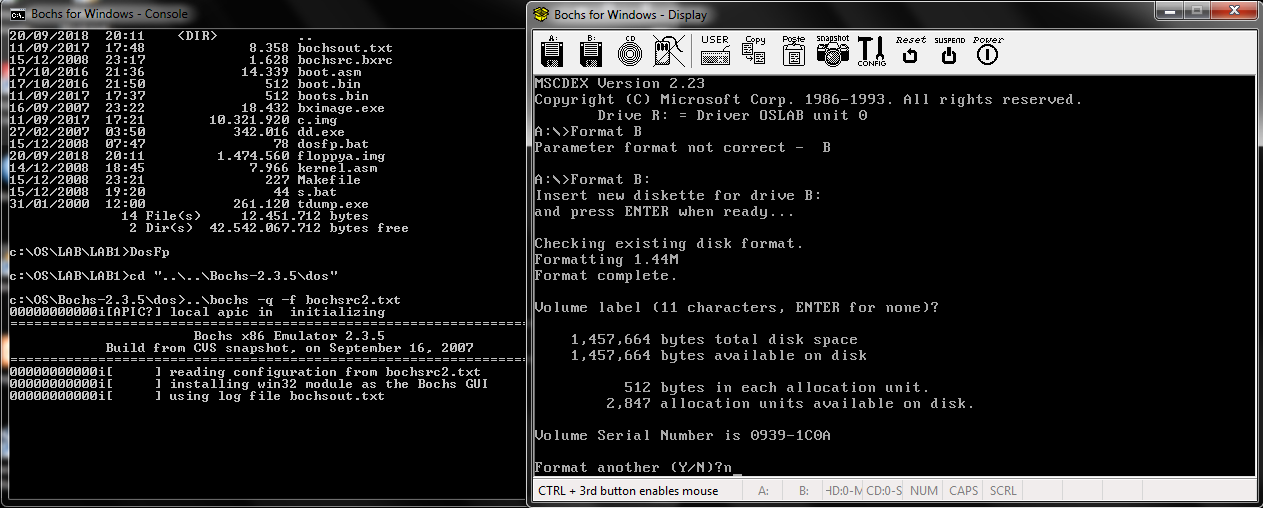
1. Cek kembali isi direktori LAB1



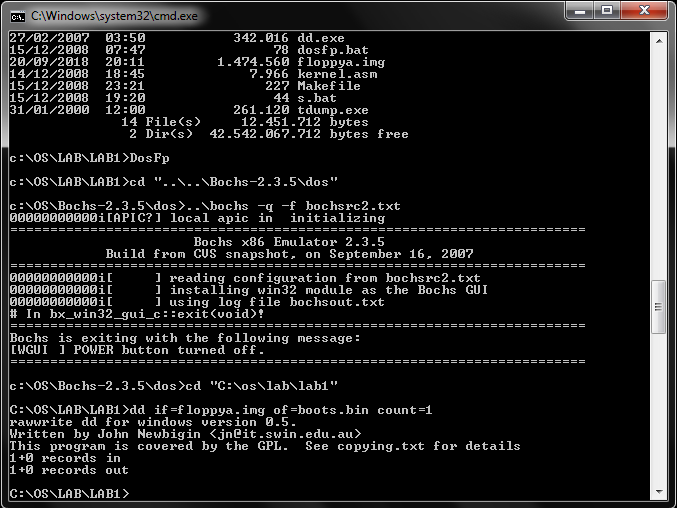
1. Kemudian format floppya.img agar menjadi bootdisk, jalankan bosc dengan mengetikan **DosFp**

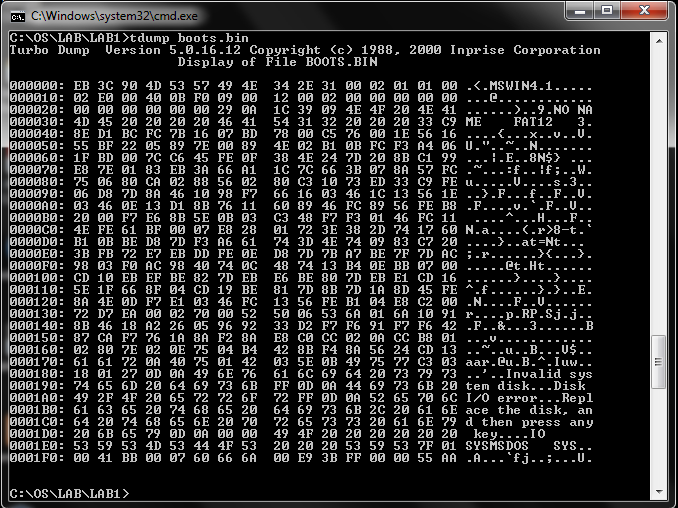


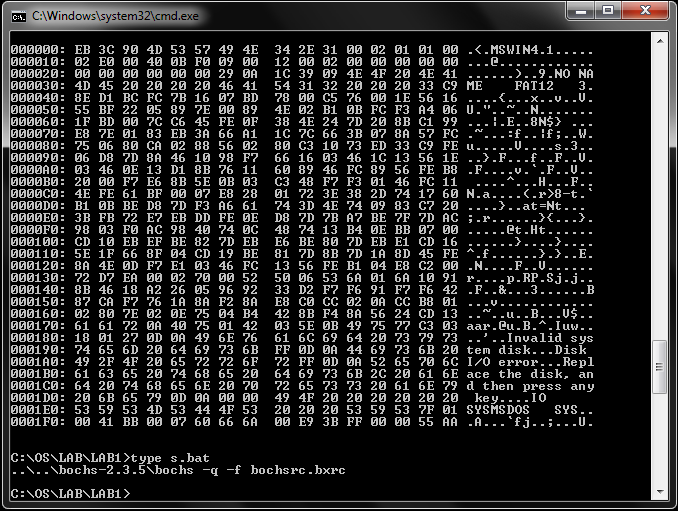
Lakukan format hingga selesai, setelah itu keluar dari bochs dengan menekan tombol power



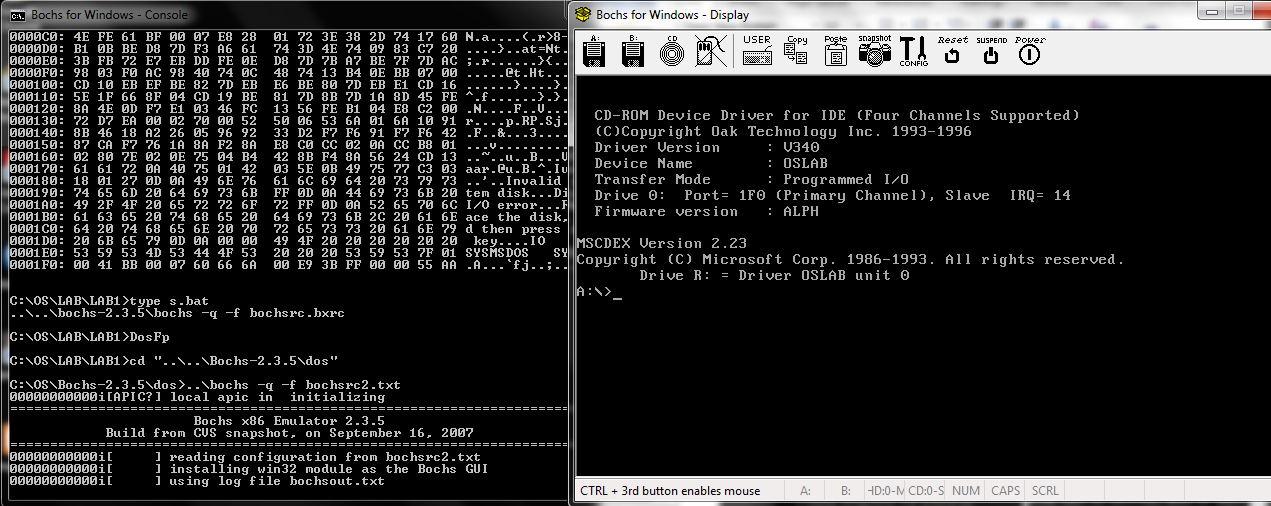
1. Salin byte data dari file “floppya.img” ke dalam “boots.bin” dengan cara mengetikkan **dd if=floppya.img of=boots.bin count=1**

****

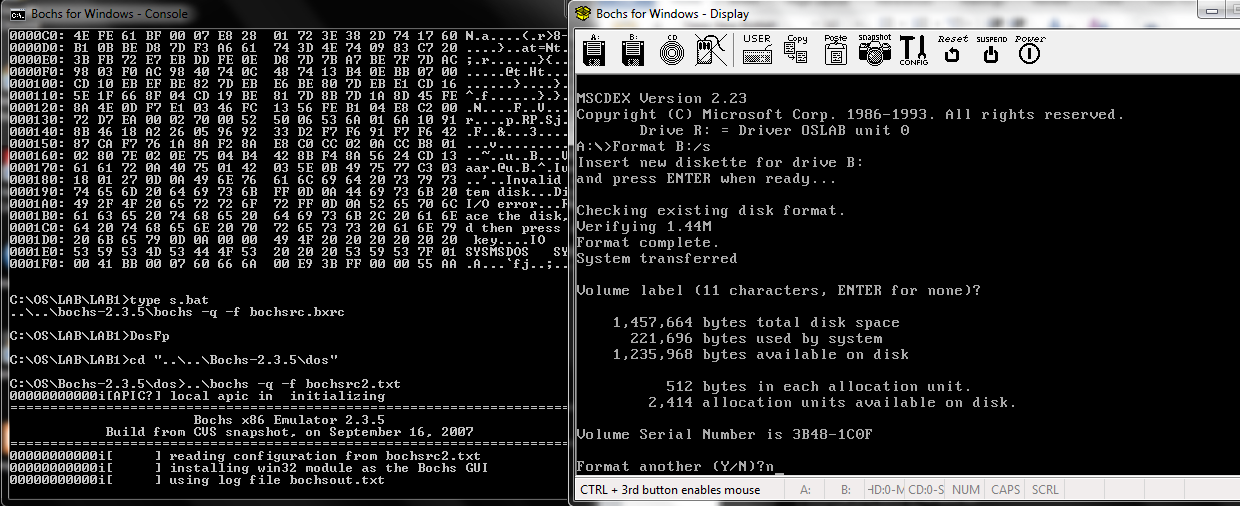
1. Melihat isi memori dari “boots.bin” dengan mengetikkan **tdump boots.bin**
2. Melihat isi file “s.bat” dengan mengetikan **type s.bat**

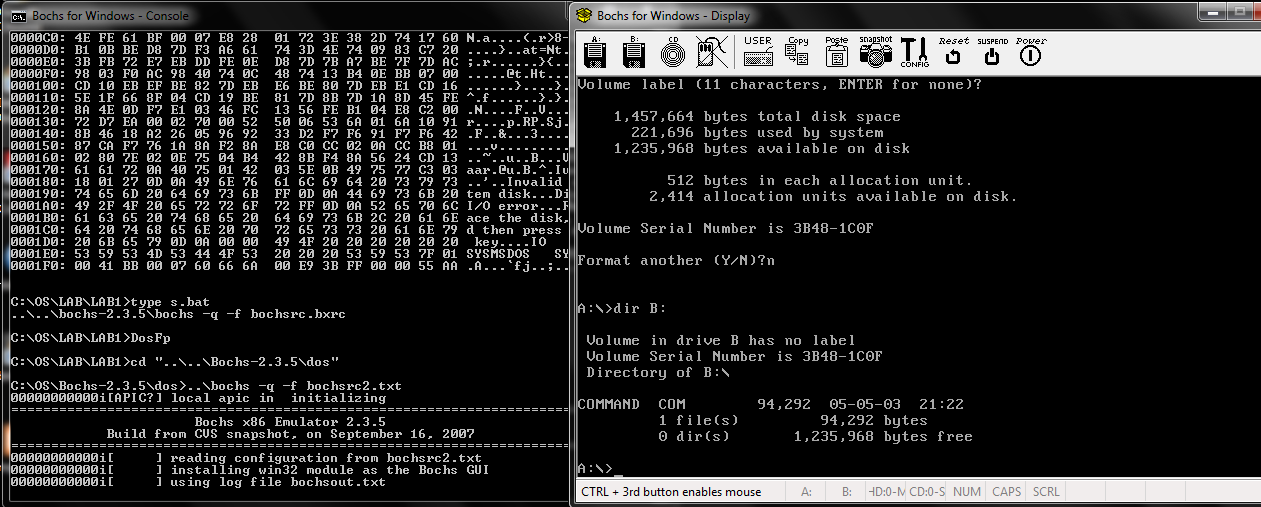
****

1. Format floppya.img, dan menambahkan sistem file ke dalamnya, dengan bantuan bosch. Ketikkan **DosFp**

****

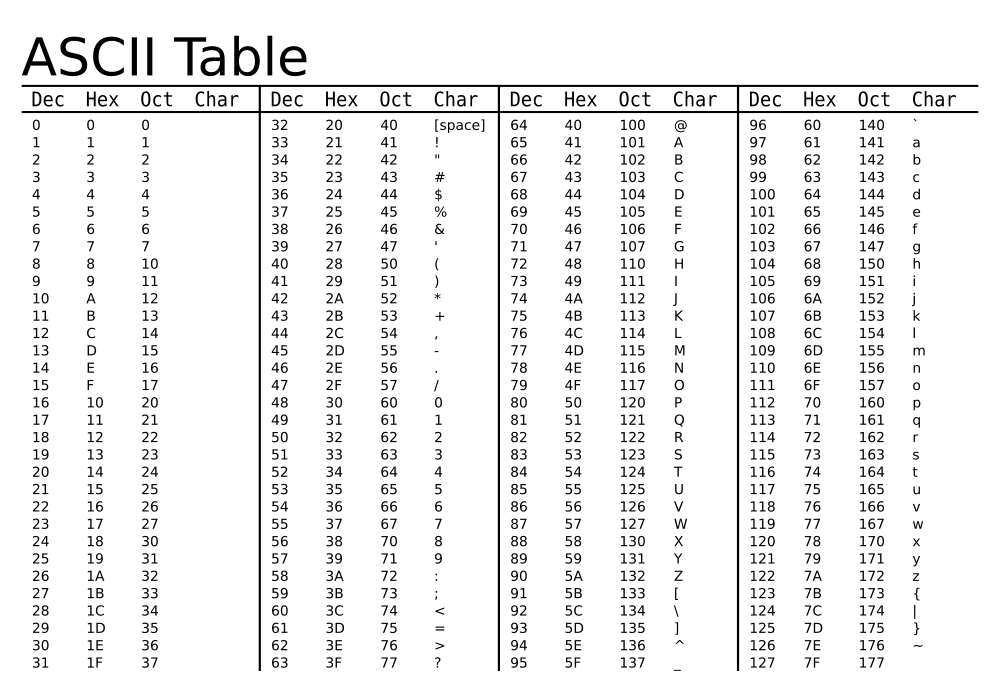
Selesaikan proses format



1. Lihat isi direktori

**Tugas**

1. Kode ASCII adalah merupakan kode standar yang digunakan dalam pertukaran informasi pada Komputer. Kode ASCII ini seperti Hex dan Unicode tetapi ASCII lebih bersifat universal.



1. Perintah Bahasa assembly untuk keluarga intel x86

* Komentar: Komentar diawali dengan tanda titik koma (;). ; ini adalah komentar
* Label : Label diakhiri dengan tanda titik dua (:). Contoh: main: ,loop: ,proses: ,keluar:
* Definisi data: **DB** : define bytes. **DW** : define words. **DD** : define double words. **EQU** : equals. Membentuk konstanta.
* Perpindahan data **MOV** : move. **LEA** : load effective address. Mengisi suatu register dengan alamat offset sebuah data. **CHG** : exchange. Menukar dua buah register langsung.
* Operasi logika: **AND** : melakukan bitwise and. sintaks: **OR** : melakukan bitwise or. **NOT** : melakukan bitwise not (*one’s complement*). **XOR** : melakukan bitwise eksklusif or. **SHL** : shift left. Menggeser bit ke kiri. Bit paling kanan diisi nol. **HR**: shift right. Menggeser bit ke kanan. Bit paling kiri diisi nol. **ROL** : rotate left. Memutar bit ke kiri. Bit paling kiri jadi paling kanan kali ini. **ROR** : rotate right. Memutar bit ke kanan. Bit paling kanan jadi paling kiri.
* Operasi matematika: **ADD** : add. Menjumlahkan dua buah register. **ADC** : add with carry. Menjumlahkan dua register dan carry flag (CF). **INC** : increment. Menjumlah isi sebuah register dengan 1. **BB** : substract with borrow. Mengurangkan dua register dan carry flag (CF). **DEC** : decrement. Mengurang isi sebuah register dengan 1. **MUL** : multiply. Mengalikan register dengan AX atau AH. **UL** : signed multiply. **DIV** : divide. Membagi AX atau DX:AX dengan sebuah register. **IDIV** : signed divide. Sama dengan DIV, hanya saja IDIV menganggap bit-bit yang ada di register sumber sudah dalam bentuk *two’s complement*. **NEG** : negate. Membuat isi register menjadi negatif (*two’s complement*).
* Pengulangan: **LOOP** : loop. Mengulang sebuah proses. Pertama register CX dikurangi satu. Bila CX sama dengan nol, maka looping berhenti. Bila tidak nol, maka lompat ke label tujuan. **LOOPE** : loop while equal. Melakukan pengulangan selama CX ≠ 0 dan ZF = 1. CX tetap dikurangi 1 sebelum diperiksa. sintaks: **LOOPZ** : loop while zero. Identik dengan LOOPE. **LOOPNE** : loop while not equal. Melakukan pengulangan selama CX ≠ 0 dan ZF = 0. CX tetap dikurangi 1 sebelum diperiksa. **LOOPNZ** : loop while not zero. Identik dengan LOOPNE. **REP** : repeat. Mengulang perintah sebanyak CX kali.