## **CSE 312 - HW3**

## Yunus ÇEVİK 141044080

Ödevin çalışması için makefile dosyasının çalıştırılması gerekmektedir. Bu dosya içerisi aşağıdaki gibidir.

CC=g++

CFLAGS=-I. --std=c++11

DEPS = gtuos.h memory.h 8080emuCPP.h memoryBase.h gtuos.cpp 8080emu.cpp

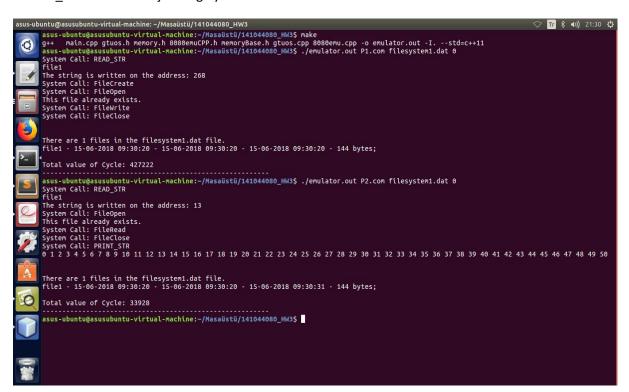
OBJ = emulator.o

buildemulator:

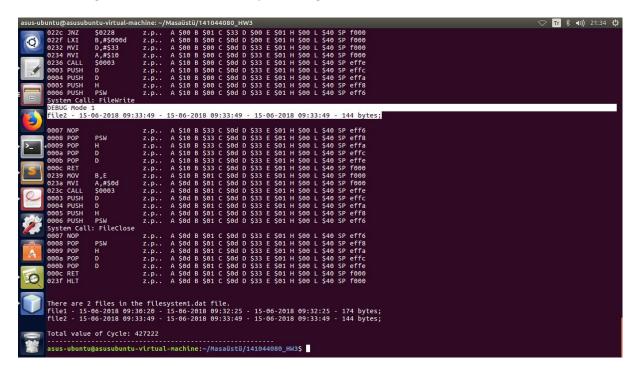
\$(CC) main.cpp \$(DEPS) -o emulator.out \$(CFLAGS)

Terminal üzerinden "make" yazıldıktan sonra "./emulator.out P1.com filesystem1.dat 0" komut satırını girerek programımızı çalıştırabiliriz. Bu komut debug 0 modun da sadece "filesystem1.dat" dosyası içerisine bizim belirlediğimiz dosya adında bir sanal dosya açıp içerisine 0 dan 50 ye kadar sayıları yazmakta ve dosya bilgilerini bulundurmaktadır.

Ayrıca "./emulator.out P2.com filesystem1.dat 0" debug 0 modun da sadece "filesystem1.dat" dosyası içerisinde bulunan bir dosyayı "FileOpen" kullanarak açarız ve içerisinde bulunan değerleri "PRINT\_STR" ile ekrana çıktı sağlayabiliriz.

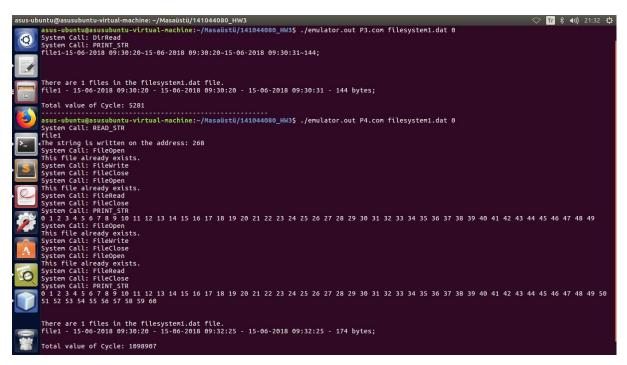


"./emulator.out P1.com filesystem1.dat 1" debug 1 modun da ise her "system call" dan sonra ekrana file table içerinde bulunan tüm dosyalardan açık olanları sadece ekrana çıktıyı gösteren moddur. Program bitiminde ise tüm dosyaların bilgileri ekrana basılır.



"./emulator.out P3.com filesystem1.dat 0" "filesystem1.dat" dosyası içerisindeki dosyaların bilgilerini ekrana yazdırmaya yarar. Bunun için "DirRead" ve "PRINT\_STR" asm içerinde komutlarını kullanmak yeterlidir.

Ayrıca "./emulator.out P4.com filesystem1.dat 0" "filesystem1.dat" dosyası içerinde yer alan bir dosya ismini girdiğimiz takdirde dosyanın contentinde bulunan değerlerin sonuna 10 tane daha sayı eklemektedir. Aşağıda bu gösterim mevcuttur.



## Yukarıdaki resimde

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 kadar sayılar aralarında boşluklu bir şekilde BC addrese gönderdiğimden dolayı ilk başta dosya **144 bytes** olarak geçmektedir. Daha sonra ise **174 bytes** olarak görünmesinin sebebi 49 sayısından sonra karakter karakter boşluk bir şekilde 60'a kadar sayıların yazılmasından kaynaklanmaktadır.

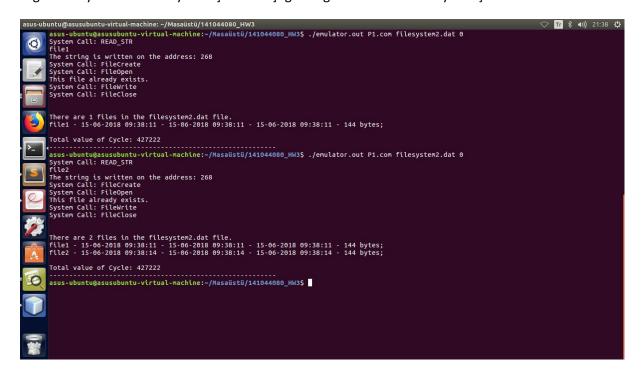
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 6 57 58 59 60

Not1: Yukarıda belirttiğim komutlar sadece "filesystem1.dat" dosyası içindir. Ayrıca gönderdiklerim içerinde "filesystem1.dat", "filesystem2.dat", "filesystem3.dat", "filesystem4.dat" ve "filesystem5.dat" dosyaları ve asm dosyalarından "P1.asm", "P2.asm", "P3.asm" ve "P4.asm" dosyalarıda mevcuttur. Ayrıca tüm filesystem dosyalarının içerisi doludur "DirRead" in olduğu "P3.com" lu denerseniz içerisinde hangi dosyaların olduğunu görebilirsiniz ve ona göre test edebilirsiniz.

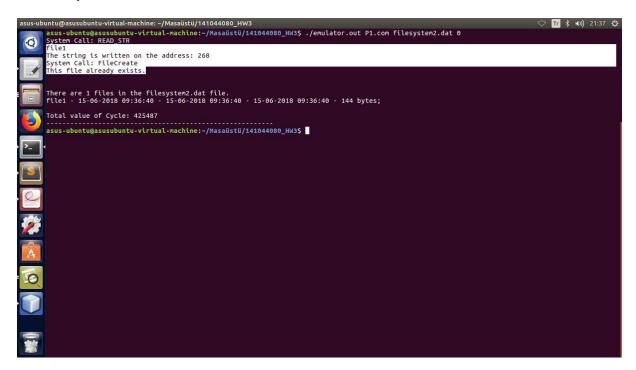
**Not2:** Bütün "filesystem.dat" dosyaları 1MB olarak sabitlenmiş olup 256KB'lık bloklar şeklinde yer almaktadır. Dosyanın özelliğinden bunu görebilirsiniz. "filesystem.dat" dosyalarını Sublime Text gibi bir editörde açtığınız taktirde bloklu yapıyı da görebilirsiniz.

Benzer ekran görüntüleri ise Screen Shot klasörü içerisinde ve aşağıda mevcuttur.

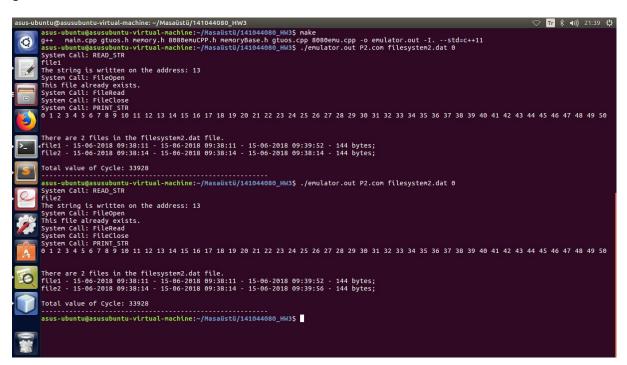
Diğer "filesystem.dat" dosyaları içerisine aşağıdaki gibi birden fazla dosya oluşturulabilir.



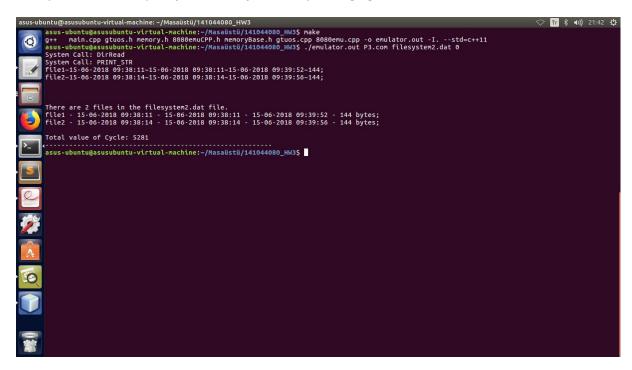
Önceden "filesystem2.dat" dosyası içerisinde oluşturduğumuz "file1" dosyasının olduğunu bildiren hata mesajı.



"filesystem2.dat" dosyası içerisine oluşturduğumuz dosyaların içerisine yazılmış 0 – 50 arası sayılar görülmektedir.



"filesystem2.dat" dosyası içerisinde kaç tane dosya olduğu görünmektedir.



"filesystem2.dat" dosyası içerisinde yer alan "file1" dosyasının contentinde bulunan 0 – 50 arası sayıların peşine 51 – 60 a kadar olan sayıların append edildiği gösterilmiştir.

