

LAPORAN TUGAS BESAR 1 MILESTONE 2
IF2230/Sistem Operasi

Pembuatan *File System* Hirarkis dan *Shell*



Dipersiapkan oleh:

Aidil Rezjki Suljztan S. 13517070

M. Khairul Makirin 13517088

Edward Alexander Jaya 13517115

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

Langkah Pengerjaan

1. Mengubah Struktur Sistem Operasi

- a. Mengubah struktur *file system* menjadi *hierarchical file system*

Berbeda dengan struktur *file system* pada Milestone 1 sebelumnya, kali ini, *file system* terbagi menjadi beberapa sektor, yaitu:

1. Map, dipindahkan ke sektor 0x100 agar sektor-sektor utilitas tidak memakan tempat yang dapat digunakan oleh *file system*
2. Dirs, pada sektor 0x101, untuk menyimpan nama semua direktori dan indeks *parent directorynya*. Dirs hanya dapat menyimpan 512 byte (15 byte untuk nama direktori, dengan byte pertama dari tiap baris (32 baris) menunjukkan indeks *parent directorynya*) jika parentnya root maka indeks *parent directorynya* 0xFF.
3. Files, pada sektor 0x102, untuk menyimpan nama file beserta indeks *parent directorynya*, yang menunjukkan file tersebut ada di direktori mana, sama seperti Dirs, Files hanya dapat menyimpan 15 byte nama file, dan maksimal file yang dapat ditampung adalah 32 file.
4. Sectors, pada sektor 0x103, untuk menyimpan sektor-sektor yang dipakai oleh suatu file, suatu file maksimal memakai 16 sektor, Files dan Sector berkorespondensi satu sama lain, yaitu baris pada Sector menunjukkan file yang sama di Files, begitupun sebaliknya.

- b. Mengubah fungsi `handleinterrupt21`

Karena akan keperluan beberapa syscall yang berubah, parameter `handleinterrupt21` juga akan diubah untuk memenuhi itu.

```
void handleInterrupt21 (int AX, int BX, int CX, int DX) {
    char AL, AH;
    AL = (char) (AX);
    AH = (char) (AX >> 8);

    switch (AL) {
        case 0x00:
            printString(BX);
            break;
        case 0x01:
            readString(BX);
            break;
        case 0x02:
            readSector(BX, CX);
            break;
    }
```

```
case 0x03:
    writeSector(BX, CX);
    break;
case 0x04:
    readFile(BX, CX, DX, AH);
    break;
case 0x05:
    writeFile(BX, CX, DX, AH);
    break;
case 0x06:
    executeProgram(BX, CX, DX, AH);
    break;
case 0x07:
    terminateProgram(BX);
    break;
case 0x08:
    makeDirectory(BX, CX, AH);
    break;
case 0x09:
    deleteFile(BX, CX, AH);
    break;
case 0x0A:
    deleteDirectory(BX, CX, AH);
    break;
case 0x20:
    putArgs(BX, CX);
    break;
case 0x21:
    getCurdir(BX);
    break;
case 0x22:
    getArgc(BX);
    break;
case 0x23:
    getArgv(BX, CX);
    break;
default:
    printString("Invalid interrupt");
```

```
}  
}
```

c. Mengubah implementasi syscall readFile

Hal-hal yang perlu diubah dari readFile sebelumnya adalah, pembacaan nama direktori dilakukan di Dirs, nama file di Files, dan sektor files di Sectors.

d. Mengubah implementasi syscall writeFile

Karena bentuk file system sekarang hirarkis, untuk dapat menulis file kita perlu memerhatikan beberapa hal yaitu:

1. Apakah terdapat n-sektor kosong di Map yang dapat menampung file
2. Apakah masih terdapat entri kosong di Files
3. Apakah direktori-direktori di path valid

Jika hal-hal diatas dapat dipenuhi, maka file dapat dituliskan ke system, caranya adalah dengan,

1. Mengambil indeks dari nama direktori terakhir di path, jadikan indeks tersebut sebagai parent directory dari file yang ingin ditulis,
2. Menuliskan nama file di entri yang kosong di Files,
3. Mencatat sector-sector yang akan diperlukan oleh file di Sectors,
4. Menuliskan isi file ke sektor-sektor yang diperlukan.

e. Mengubah implementasi syscall executeProgram

ExecuteProgram harus diubah agar dapat menerima path tempat program, untuk penlokasian filenya sendiri sudah ditangani oleh readFile, sehingga exucuteProgram tidak banyak berubah.

f. Mengimplementasi syscall terminateProgram

TerminateProgram akan berjalan setiap selesai executeProgram agar process dapat kembali ke *Shell* lagi.

g. Mengimplementasi syscall makeDirectory

MakeDirectory akan membuat directory baru jika,

1. Path baru valid, dan
2. Terdapat entri kosong pada Dirs.

Jika kedua syarat di atas terpenuhi, makeDirectory akan menuliskan nama direktori baru di Dirs dengan parent indeksinya adalah indeks directory dari nama directory terakhir di path.

h. Mengimplementasi syscall deleteFile

DeleteFile akan menghapus file dari system jika path yang diberikan valid, kemudian untuk menghapus sebuah file, pertama hapus semua isi dari sektor-sektor yang tadinya dipakai oleh file tersebut, kemudian ubah ketersediaan sektor-sektor tersebut di Map, kemudian hapus entri di Files dan Sectors yang menunjuk pada file yang ingin dihapus.

- i. Mengimplementasi syscall deleteDirectory

DeleteDirectory akan menghapus directory yang dimaksud pada path dan menghapus secara rekursif direktori beserta file-file di dalam direktori tersebut.

- j. Mengimplementasi syscall putArgs, getArgc, getArgv

2. Membuat *Shell* Sistem Operasi

Shell dalam sistem operasi akan diimplementasikan sebagai program bernama shell.c, dan setelah sistem operasi booting, program shell.c akan dieksekusi melalui *syscall* executeProgram.

- a. Implementasi perintah cd

Perintah cd akan mengubah *current directory* menjadi *path directory* jika path valid.

```
Bochs x86-64 emulator, http://bochs.sourceforge.net/
USER Copy Paste Snapshot Reset suspend/Power
F12
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
A: NUM CAPS SCRL
IPS: 21.087M

Plex86/Bochs VGABios (PCI) current-cvs 08 Apr 2016
This VGA/VBE Bios is released under the GNU LGPL

Please visit :
. http://bochs.sourceforge.net
. http://www.nongnu.org/vgabios

Bochs VBE Display Adapter enabled

Bochs BIOS - build: 09/02/12
$Revision: 11318 $ $Date: 2012-08-06 19:59:54 +0200 (Mo, 06. Aug 2012) $
Options: apmbios pcibios pnpbios eltorito rombios32

Press F12 for boot menu.

Booting from Floppy...
$ mkdir Pak Eko
$ cd Pak Eko
$ ls
Berikut adalah direktori pada direktori ini:

Berikut adalah files pada direktori ini:
$
```

(Gambar contoh cd)

- b. Implementasi perintah menjalankan program

Untuk menjalankan suatu program, dalam shell harus ditulis “./” diikuti dengan nama programnya.

Perintah `cd` akan mengubah *current directory* menjadi *path directory* jika path valid.

```
Bochs x86-64 emulator, http://bochs.sourceforge.net/
Plex86/Bochs UGABios (PCI) current-cvs 08 Apr 2016
This UGA/UBE Bios is released under the GNU LGPL

Please visit :
. http://bochs.sourceforge.net
. http://www.nongnu.org/vgabios

Bochs UBE Display Adapter enabled

Bochs BIOS - build: 09/02/12
$Revision: 11318 $ $Date: 2012-08-06 19:59:54 +0200 (Mo, 06. Aug 2012) $
Options: apmbios pcibios pnpbios eltorito rombios32

Press F12 for boot menu.

Booting from Floppy...
$ ./keyproc2 01 02
Starting key generation procedure 2 as:
K01-Kelompok02

Created directory "K102"
Created directory "K102/in"
Please create a file containing your access code at "K102/in/code.txt"
$
IPS: 20.823M | A: | NUM | CAPS | SCRL | | | | | | | |
```

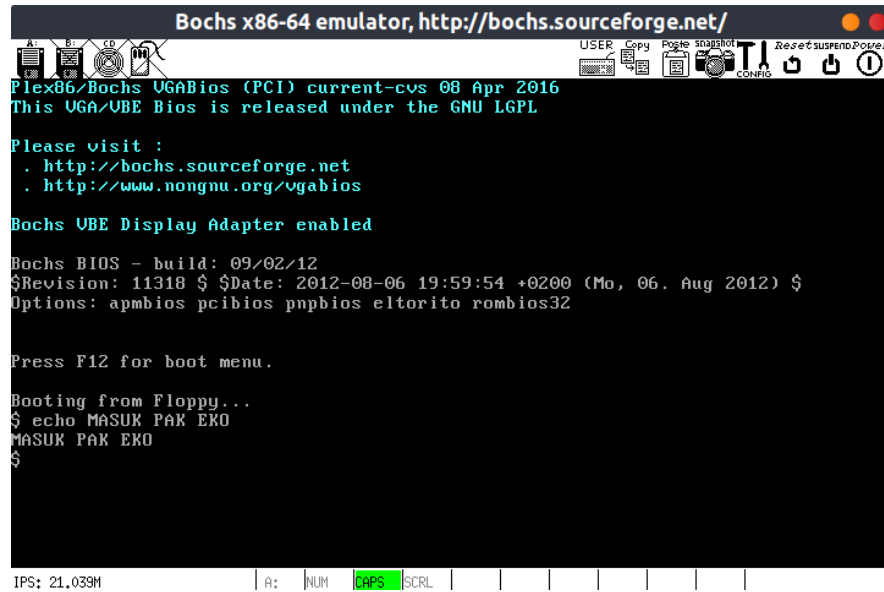
(Gambar contoh menjalankan Program)

3. Membuat Program Utilitas

Program utilitas adalah program command yang dapat dijalankan pada shell dari directory manapun, maka dari itu program utilitas harus disimpan di directory *root*.

a. Membuat program echo

Program echo akan mencetak tiap parameter yang diberikan.



The screenshot shows the Bochs x86-64 emulator window. The title bar reads "Bochs x86-64 emulator, http://bochs.sourceforge.net/". The main window displays the following text:

```
Flex86/Bochs VGABios (PCI) current-cvs 08 Apr 2016
This UGA/VE Bios is released under the GNU LGPL

Please visit :
. http://bochs.sourceforge.net
. http://www.nongnu.org/vgabios

Bochs VBE Display Adapter enabled

Bochs BIOS - build: 09/02/12
$Revision: 11318 $ $Date: 2012-08-06 19:59:54 +0200 (Mo, 06. Aug 2012) $
Options: apmbios pcibios pnpbios eltorito rombios32

Press F12 for boot menu.

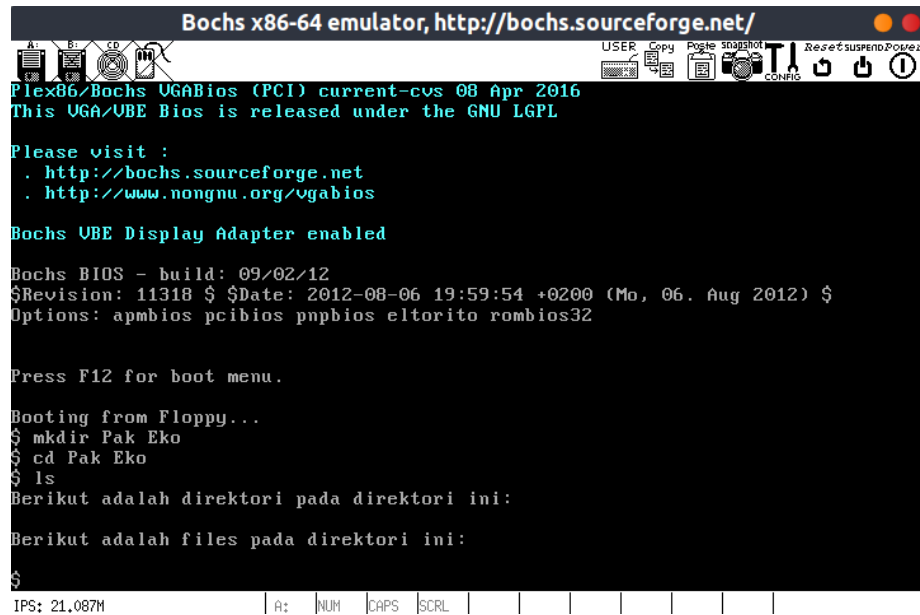
Booting from Floppy...
$ echo MASUK PAK EKO
MASUK PAK EKO
$
```

The status bar at the bottom shows "IPS: 21.039M" and a keyboard layout indicator with "A:", "NUM", "CAPS", and "SCRL" buttons.

(Gambar contoh echo)

b. Membuat program mkdir

Program mkdir akan membuat suatu direktori baru di bawah direktori sekarang, dengan nama direktorinya sesuai dengan parameter yang diberikan (hanya 1 parameter).



The screenshot shows the Bochs x86-64 emulator window. The title bar reads "Bochs x86-64 emulator, http://bochs.sourceforge.net/". The main window displays the following text:

```
Flex86/Bochs VGABios (PCI) current-cvs 08 Apr 2016
This UGA/VE Bios is released under the GNU LGPL

Please visit :
. http://bochs.sourceforge.net
. http://www.nongnu.org/vgabios

Bochs VBE Display Adapter enabled

Bochs BIOS - build: 09/02/12
$Revision: 11318 $ $Date: 2012-08-06 19:59:54 +0200 (Mo, 06. Aug 2012) $
Options: apmbios pcibios pnpbios eltorito rombios32

Press F12 for boot menu.

Booting from Floppy...
$ mkdir Pak Eko
$ cd Pak Eko
$ ls
Berikut adalah direktori pada direktori ini:

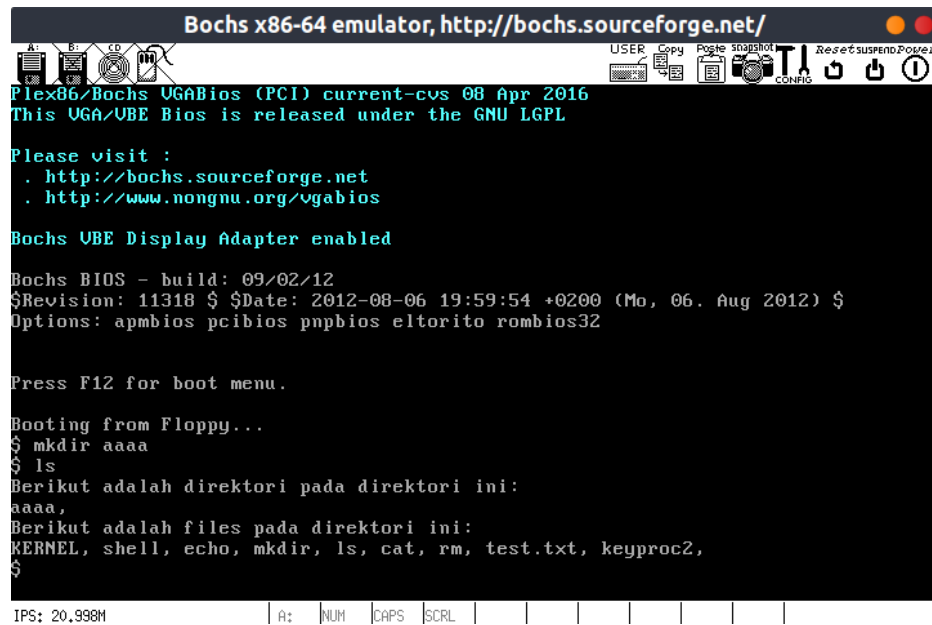
Berikut adalah files pada direktori ini:
$
```

The status bar at the bottom shows "IPS: 21.087M" and a keyboard layout indicator with "A:", "NUM", "CAPS", and "SCRL" buttons.

(Gambar contoh mkdir)

c. Membuat program ls

Program ls akan menampilkan seluruh file dan folder yang terdapat pada direktori saat ini.



```
Bochs x86-64 emulator, http://bochs.sourceforge.net/
Plex86/Bochs UGABios (PCI) current-cvs 08 Apr 2016
This UGA/VE Bios is released under the GNU LGPL

Please visit :
. http://bochs.sourceforge.net
. http://www.nongnu.org/vgabios

Bochs VBE Display Adapter enabled

Bochs BIOS - build: 09/02/12
$Revision: 11318 $ $Date: 2012-08-06 19:59:54 +0200 (Mo, 06. Aug 2012) $
Options: apmbios pcibios pnpbios eltorito rombios32

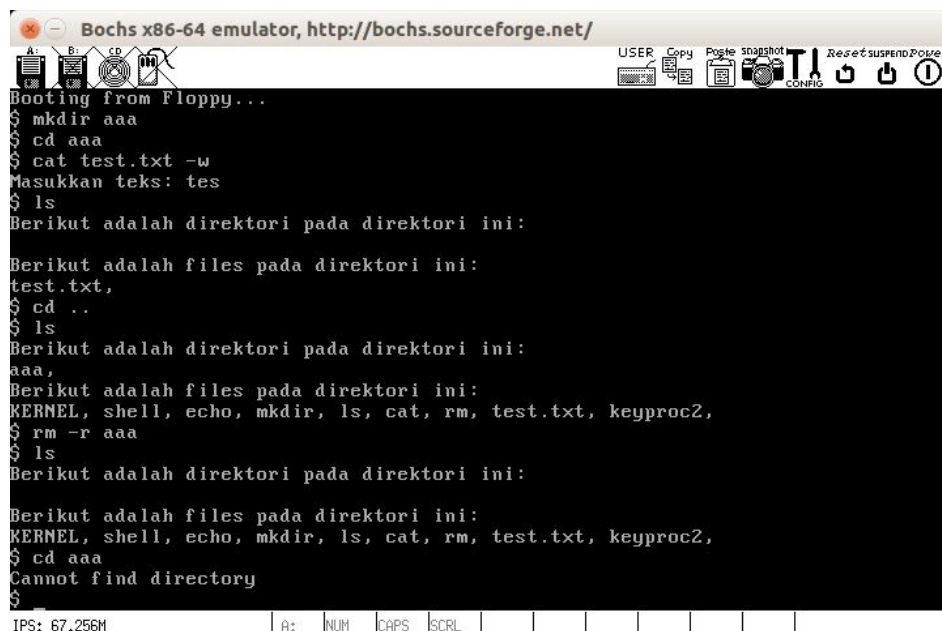
Press F12 for boot menu.

Booting from Floppy...
$ mkdir aaaa
$ ls
Berikut adalah direktori pada direktori ini:
aaaa,
Berikut adalah files pada direktori ini:
KERNEL, shell, echo, mkdir, ls, cat, rm, test.txt, keyproc2,
$
```

(Gambar contoh ls)

d. Membuat program rm

Program rm akan menghapus folder dan/atau file pada direktori path yang diberikan sebagai argumen



```
Bochs x86-64 emulator, http://bochs.sourceforge.net/
Booting from Floppy...
$ mkdir aaa
$ cd aaa
$ cat test.txt -w
Masukkan teks: tes
$ ls
Berikut adalah direktori pada direktori ini:

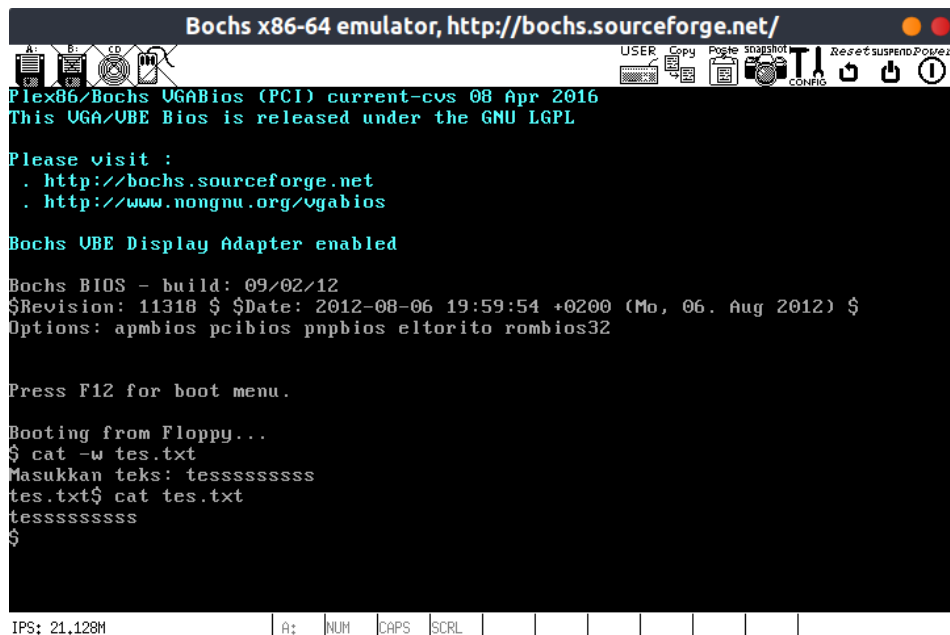
Berikut adalah files pada direktori ini:
test.txt,
$ cd ..
$ ls
Berikut adalah direktori pada direktori ini:
aaa,
Berikut adalah files pada direktori ini:
KERNEL, shell, echo, mkdir, ls, cat, rm, test.txt, keyproc2,
$ rm -r aaa
$ ls
Berikut adalah direktori pada direktori ini:

Berikut adalah files pada direktori ini:
KERNEL, shell, echo, mkdir, ls, cat, rm, test.txt, keyproc2,
$ cd aaa
Cannot find directory
$
```

(Gambar contoh rm)

e. Membuat program cat

Program cat mempunyai 2 mode, mode menulis dan mode mencetak, jika ingin mode menulis cat harus disertai dengan argumen **-w** diikuti dengan path dari file yang mau ditulis, jika mencetak tidak perlu pakai flag **-w**, tetapi hanya path dari file yang mau dicetak.



```
Bochs x86-64 emulator, http://bochs.sourceforge.net/
F1ex86/Bochs VGABios (PCI) current-cvs 08 Apr 2016
This VGA/UBE Bios is released under the GNU LGPL

Please visit :
. http://bochs.sourceforge.net
. http://www.nongnu.org/vgabios

Bochs UBE Display Adapter enabled

Bochs BIOS - build: 09/02/12
$Revision: 11318 $ $Date: 2012-08-06 19:59:54 +0200 (Mo, 06. Aug 2012) $
Options: apmbios pcibios pnpbios eltorito rombios32

Press F12 for boot menu.

Booting from Floppy...
$ cat -w tes.txt
Masukkan teks: tssssssssss
tes.txt$ cat tes.txt
tssssssssss
$
```

(Gambar contoh cat)

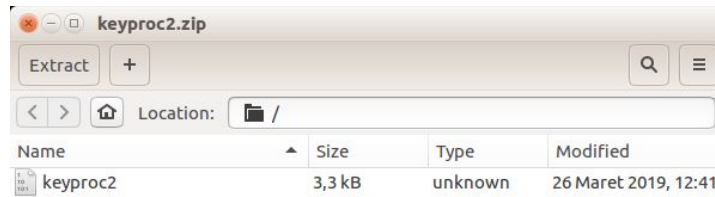
4. Menjalankan program pengujian

Untuk menjalankan program pengujian, akan dibutuhkan :

1. Fungsi readFile
2. Fungsi writeFile
3. Fungsi executeProgram
4. Fungsi terminateProgram
5. Fungsi makeDirectory
6. Shell sistem operasi
7. Program utilitas ls
8. Program utilitas cat

Langkah-langkahnya adalah sebagai berikut :

1. Download *archive.zip* dari milis



(Gambar keyproc2)

2. Masukkan “**keyproc2**” yang ada di zip tersebut ke root directory dengan perintah

```
./loadFile keyproc2
```

3. Pastikan **keyproc2** sudah ada di root dengan menjalankan program **ls**
4. Jalankan **keyproc2** dengan parameter nomor kelas dan nomor kelompok
5. Lakukan **cat -w** untuk menulis access code
6. Jalankan kembali **keyproc2**
7. Gunakan **cat** untuk membaca file yang dihasilkan **keyproc2**

```
Bochs x86-64 emulator, http://bochs.sourceforge.net/
Press F12 for boot menu.
Booting from Floppy...
$ ./keyproc2 01 02
Starting key generation procedure 2 as:
  K01-Kelompok02

Created directory "K102"
Created directory "K102/in"
Please create a file containing your access code at "K102/in/code.txt"
$ cat -w K102/in/code.txt
Masukkan teks: sEQEvfBo
K102/in/code.txt$ cat K102/in/code.txt
sEQEvfBo
$ ./keyproc2 01 02
Starting key generation procedure 2 as:
  K01-Kelompok02

Read file "K102/in/code.txt"
Created directory "K102/out"
Created file K102/out/key.txt
$ cat K102/out/key.txt
ZKtD28L
$
```

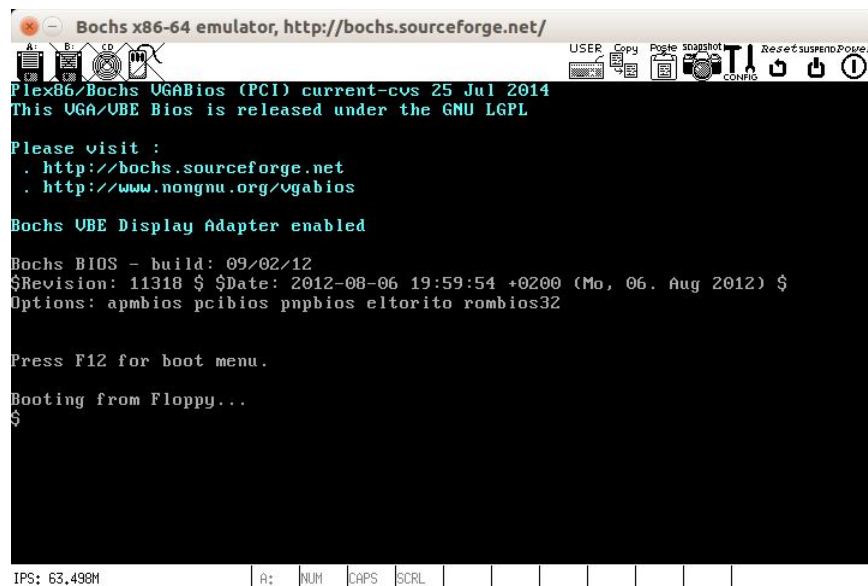
(Gambar contoh menjalankan keyproc2)

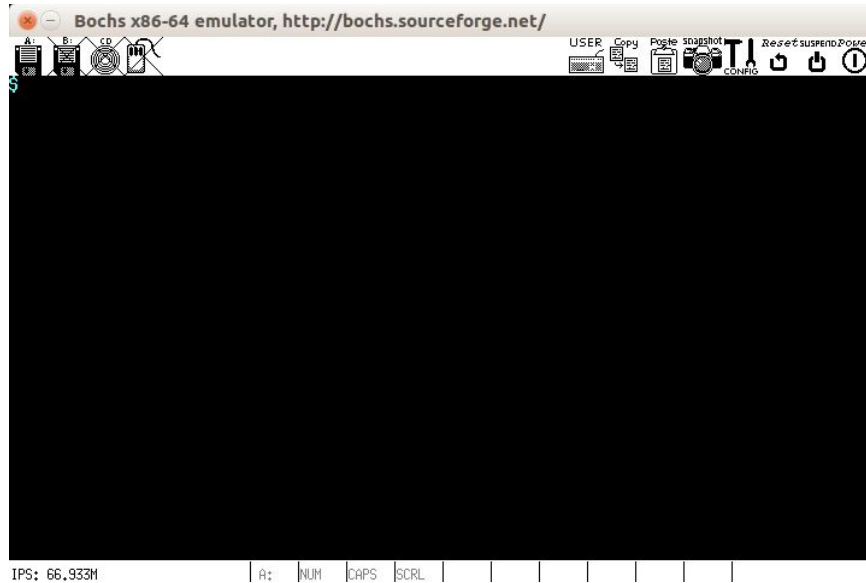
5. BONUS

a. Implementasi clr

Perintah clr akan menghapus layar console dari teks yang tercetak, diimplementasikan sebagai berikut,

```
void clearScreen(int _lines){  
    //Prosedur membersihkan layar sebanyak _lines relatif terhadap  
    0x8000.  
    int i;  
    //Membersihkan layar.  
    for (i = 0; i < 80 * _lines; i++){  
        putInMemory(0xB000, 0x8000 + (i * 2), ' ');  
    }  
    interrupt(0x10, 0x0200, 0, 0, 0);  
}
```





b. Implementasi mv

Perintah mv akan dapat memindahkan file ke folder di direktorinya, file ke parent foldernya, dan renaming file, implementasinya seperti berikut

```
void move(char *path1, char *path2, char curDir){
    int id1, id2, i, j, k;
    int isFolder1 = 0;
    int isFile2 = 0;
    int isToRoot = 0;
    int loop = 1;
    char files[512];
    char dirs[512];
    char namaBaru[MAX_FILENAME];

    readSector(files, FILES_SECTOR);
    readSector(dirs, DIRS_SECTOR);

    id1 = findIndexFile(path1, curDir);
    if (id1 == -1) {
        id1 = findIndexDirectory(path1, curDir);
        isFolder1 = 1;
    }

    if (path2[0] == '.' && path2[1] == '.'){
        id2 = findIndexFile(path1, curDir);
        id2 = dirs[files[id2 * 16] * 16];
    }
```

```

        isToRoot = 1;
    }

    if (!isToRoot){
        id2 = findIndexDirectory(path2, curDir);
        if (id2 == -1){
            id2 = findIndexFile(path2, curDir);
            isFile2 = 1;
        }
    }

    if (id1 != -1 && id2 != -1){
        if (!isFolder1 && !isFile2){
            files [id1 * 16] = id2;
        }else if (isFolder1 && !isFile2) {
            dirs [id1 * 16] = id2;
        }
    }else if (!isFolder1 && isFile2 && id2 == -1 && id1 != -1){
        for (i = 1; i < 16; ++i){
            files[id1*16 + i] = path2[i - 1];
        }
    }

    writeSector(files, FILES_SECTOR);
    writeSector(dirs, DIRS_SECTOR);
}

```

```

Bochs x86-64 emulator, http://bochs.sourceforge.net/
$ ls
Berikut adalah direktori pada direktori ini:
aaa,
Berikut adalah files pada direktori ini:
KERNEL, shell, echo, mkdir, ls, cat, rm, test.txt, keyproc2, clr.o,
$ cd aaa
$ cat bb.txt -w
Masukkan teks: hehe
$ mv bb.txt ..
$ ls
Berikut adalah direktori pada direktori ini:
Berikut adalah files pada direktori ini:
$ cd ..
$ ls
Berikut adalah direktori pada direktori ini:
aaa,
Berikut adalah files pada direktori ini:
KERNEL, shell, echo, mkdir, ls, cat, rm, test.txt, keyproc2, clr.o, bb.txt,
$

```

IPS: 68,327M | A: | NUM | CAPS | SCRL | | | | | | | |

Pembagian Tugas

NIM	Nama	Apa yang dikerjakan	Persentase kontribusi
13517070	Aidil Rezki Suljztan S.	Main, Fungsi Syscal, Program Utilitas, Debugging, Bonus	33,3%
13517088	M. Khairul Makirin	Laporan, Fungsi Syscall, Program Utilitas, Debugging, Bonus	33,3%
13517115	Edward Alexander Jaya	Shell, Main, Debugging, Fungsi Syscall, Program Utilita, Bonus	33,3%

Kesulitan dan Feedback

Kesulitan yang kami hadapi untuk tugas kecil kali ini adalah susahnya untuk berkoordinasi satu sama lain dikarenakan jadwal kami yang ketat dan banyaknya tugas-tugas besar maupun kecil di mata kuliah lain, sehingga kami hanya sekali kumpul untuk melakukan kerja kelompok sisanya kami koordinasi lewat aplikasi *Line*. *Feedback* untuk Milestone kali ini adalah tugasnya betul-betul lebih susah daripada Milestone pertama, ditambah lagi karena *syscall* *putArgs* masih terdapat bug dan baru diberitahukan 2,5 jam sebelum deadline awalnya. Kesulitan lainnya yang kami alami adalah tentang i saat meneruskan parameter ke *argv* (jika argumennya > 1), selain itu kita tidak diberitahu secara eksplisit apa yang harus dilakukan jika kehabisan memori untuk kernel, sehingga kita harus *debugging* selama 3 jam hanya karena memori tidak cukup.