

# SCHizOS operációs rendszer bemutatása és működése

## Tartalom

SCHizOS operációs rendszer bemutatása és működése.....	1
Általános információk .....	1
Források.....	2
Használathoz szükséges eszközök .....	3
Ha magunk akarjuk fordítani a kernelt .....	3
Példa használat a QEMU emulátor segítségével .....	4
Demo futás .....	11
1. Demo: A felhasználó megkeres egy fájlt, és ki akarja iratni a tartalmát.....	11
2. Demo: A billentyűzet választás tesztelése.....	12
Taszkok kezelése.....	13
Memóriakezelés menete .....	13
A bootolás menete.....	13
1. Fázis: .....	13
2. Fázis: .....	15
Végző .....	17

## Általános információk

Készítette: Funk Gábor

A SCHizOS operációs rendszer egy 32 bites operációs rendszer mely monolitikus kernel felépítést használ. Ezen felül ráfér egy floppy lemezre is.

Főbb dolgok, amiket tartalmaz:

- 2 fokozatú bootolás
- FAT12-es fájlrendszer, mely a Kernel és még extra csomagolt adatok betöltésére szolgál
- Egy kis shell system minimális parancsokkal
- Billentyűzet nyelv megváltoztatás parancsokkal
- Random generálás pseudo módon vagy a beépített CPU hívással
- Egy taszk kezelő rendszer, mely képes perzisztens, vagy úgynevezett nem perzisztens taszkokat kezelni.

- Disk driver
- A kernel betöltésekor átvált az operációs rendszer 16-bit „Real” módból 32-bites „Protected” módba.
- Hanglejátszás támogatása
- Saját memória kezelés (több mód is van)
- Initial Ramdisk betöltése

Fontos limitációk:

- Még nincs grafikus driver, így az alap konzolra van a kinézet limitálva, mely 25 sor és 80 oszlop széles. (Ennek a drivernek a jól megírása körülbelül annyi idő lenne, mint az eddigi egész operációs rendszernek a megírása)

Későbbi tervek:

Habár nem tervezek nagyon komplex operációs rendszert csinálni a SCHizOS-ből, vannak ötleteim, amiket meg akarok csinálni:

- ATA driver, hogy a 32-bites módból is lehessen a disk tartalmát módosítani.
- Több parancs (rm, cp, mv)
- Nyelvi beállítások
- Egy saját szövegszerkesztő vagy fájl szerkesztő

## Források

Az OSDev wiki nagyon sokat segített azoknak a koncepcióknak a megértésében, amik szükségessé váltak a SCHizOS elkészüléséhez:

[https://wiki.osdev.org/Expanded\\_Main\\_Page](https://wiki.osdev.org/Expanded_Main_Page)

A FAT12 implementáció az itteni leírás alapján készítettem el.

A legtöbb segítséget ez nyújtotta nekem, ugyanis ez leírta a fontosabb dolgokat, milyen flag biteket mikor kell használni, illetve a 32-bit módba való átváltáshoz is elengedhetetlen volt.

Fontos megjegyezni, hogy a bootolás után már a működés nagy részét én találtam ki, így az lehet, hogy eltérhet a szabványos működéstől. (Például: Shell parancs kezelés, Taszk kezelés)

## Használathoz szükséges eszközök

Fontos, hogy az operációs rendszert még csak Ubuntu Linux alatt lehet építeni, én a kipróbáláshoz a QEMU emulátort ajánlom, amit Ubuntura egyszerűen, ezekkel a parancsokkal lehet letelepíteni:

```
sudo apt update  
sudo apt install qemu-system
```

Fontos, hogy nem elég a qemu-user-static, hanem a qemu-system változata kell.

Először is szükséges a forráskód és opcionálisan a bináris fájl a működés megértéséhez.

A forráskódot 2 helyről lehet letölteni:

- <https://afghangoat.hu/imports/src/SCHizOS/SCHizOS.zip>
- <https://github.com/afghangoat/SCHizOS>

Az előbbi linket jobban ajánlom, az utóbbi az egy tükör, ami igazából a Githubra tükrözi az elsőt.

## Ha magunk akarjuk fordítani a kernelt

Ez a szekció csak akkor releváns, ha nem a meglévő fordított operációs rendszert akarjuk használni, hanem sajátot akarunk forrásból fordítani.

Ekkor szükségünk lesz 3 eszközre:

- Binutils (Ajánlott verzió: 2.37)
- GCC (Ajánlott verzió: 11.1.0)
- i686-elf

Ezek a legegyszerűbben megszerezhetőek ebből a Github repositoryból:

<https://github.com/lordmilko/i686-elf-tools>

Vagy opcionálisan a GNU.org oldaláról:

<https://ftp.gnu.org/gnu/binutils/binutils-2.37.tar.xz>

<https://ftp.gnu.org/gnu/gcc/gcc-11.1.0/gcc-11.1.0.tar.xz>

Ezek építésére használható a build\_scripts/toolchain.mk script.

Ezek után a

```
make
```

parancsot kell futtatni. Ilyenkor ha jó verziójú a Binutils és GCC build, akkor ez hiba nélkül fog lefutni.

## Példa használat a QEMU emulátor segítségével

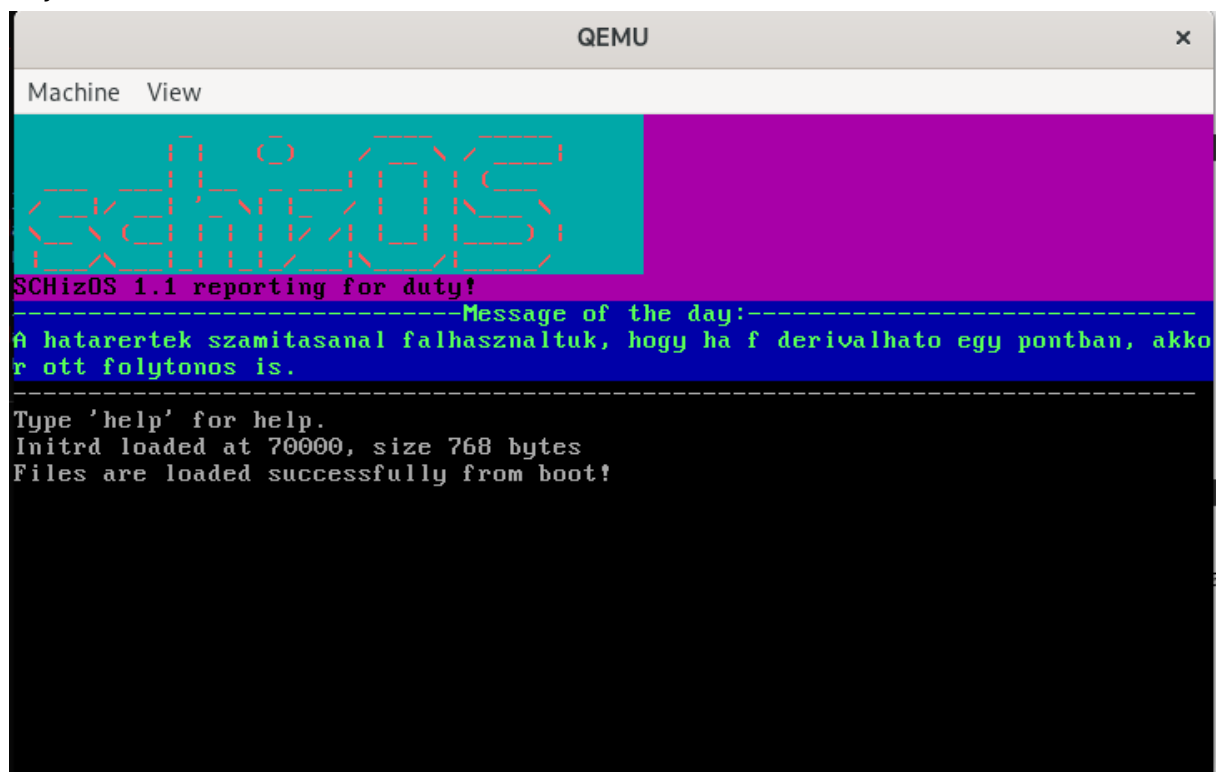
Az alábbi szekció az operációs rendszer „kommersz” használatát fogja tárgyalni.

Először is futtatni kell a run.sh scriptet a terminálba:

```
./run.sh
```

Ez a shell script a QEMU system emulátort meg fogja hívni a megfelelő beállításokkal. (Megadja, hogy a floppy image-et emulálja, és a hang hallatásához szükséges beállításokat is kérvényezi).

A parancs után megtörténik a bootolás, és betölt a kernel. A kernel betöltése után ezt látja a felhasználó:

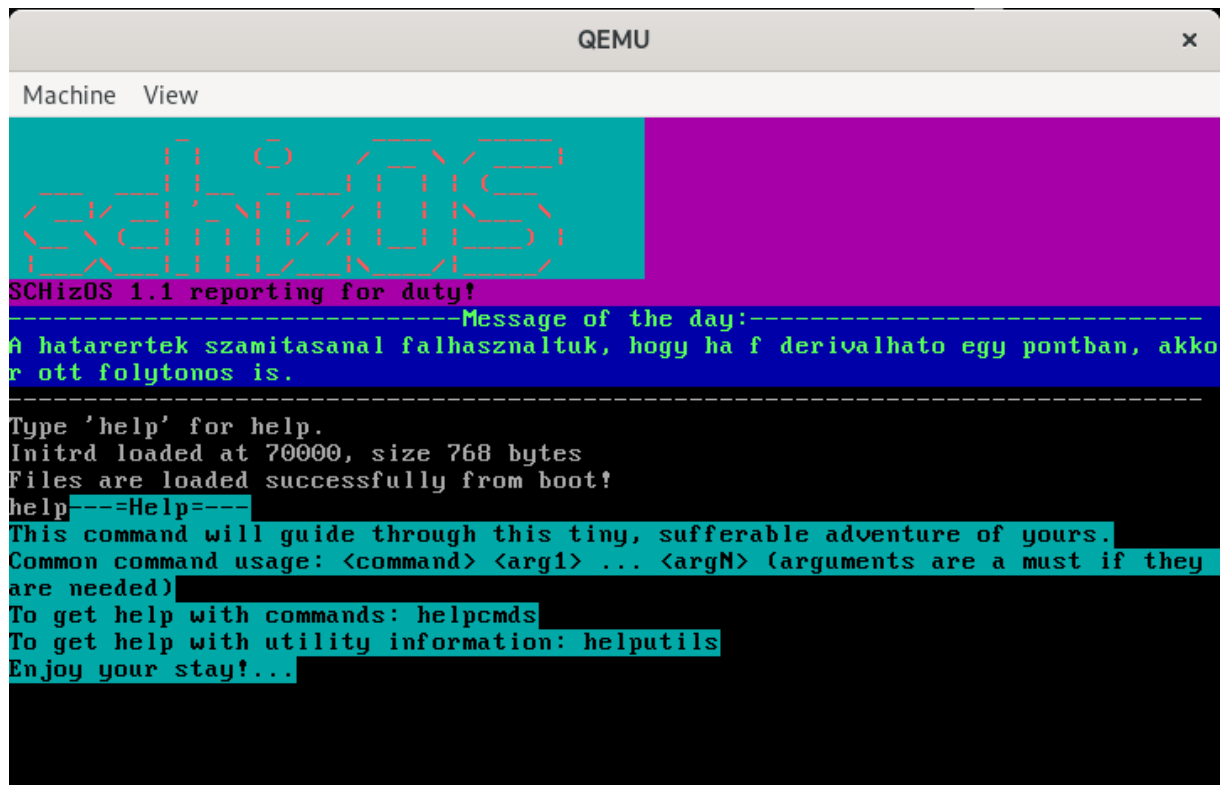


A schizOS címere is látszik, valamint a fő és minor verziója. Ezen felül látszik egy „Message of the day”, vagy motd, ez igazából arra jó, hogy véletlenszerű tippeket ad az Analízis 1 nevű tárgyhoz.

Ezen felül az operációs rendszer jelzi, hogy sikeres volt a fájlok betöltése a bootolás során, és, hogy a ramdiszk is inicializálva lett. A 70000 szám az azt jelenti, hogy a 70000.-ik memória címre. Erre majd később kitérek, a SCHizOS hibrid memória kezelést használ, ami azt jelenti, hogy van olyan eset, ahol direktbe címzi a memóriát, míg máshol virtuálisan címzi azt.

Azt is észre lehet venni, hogy a rendszer automatikus sortörést végez az olyan szövegek esetében, amik nem férnek el egy 80 karakteres sorba.

Mivel az operációs rendszer nagyon alap így majdnem minden jól van dokumentálva azon belül is. A „help” parancs az első, amit felajánl a rendszer a végfelhasználónak. Ez a parancs nem egyezik a Linux „help” paranccsal, mivel ez egyelőre csak általános információkat mond el az operációs rendszerről és abba való navigálásról.



```
QEMU
Machine View
SCHizOS 1.1 reporting for duty!
-----Message of the day:-----
A hatarertek szamitasanal falhasznaltuk, hogy ha f derivalhato egy pontban, akkor ott folytonos is.
-----
Type 'help' for help.
Initrd loaded at 700000, size 768 bytes
Files are loaded successfully from boot!
help---=Help---
This command will guide through this tiny, sufferable adventure of yours.
Common command usage: <command> <arg1> ... <argN> (arguments are a must if they are needed)
To get help with commands: helpcmds
To get help with utility information: helputils
Enjoy your stay!...
```

A help parancsot begépelve egy ciánnal jelölt üzenet jelenik meg. Ez információt ad további segítő parancsokról és arról is, hogy mi az általános parancs szintaxis. A parancs utáni ENTER gomb lenyomása után észre lehet venni, hogy amennyiben a hang be van kapcsolva az emulálás közben, egy mély „beep” hang játszódik le a parancs végrehajtása után. Ez a hang minden parancs után lejátszódik.

A következő fontos parancs a „helputils”:

```
QEMU
Machine View
SCHizOS 1.1 reporting for duty!
-----Message of the day:-----
A hatarertek szamitasanal falhasznaltuk, hogy ha f derivalhato egy pontban, akkor ott folytonos is.
-----
Type 'help' for help.
Initrd loaded at 70000, size 768 bytes
Files are loaded successfully from boot!
help---=Help=---
This command will guide through this tiny, sufferable adventure of yours.
Common command usage: <command> <arg1> ... <argN> (arguments are a must if they are needed)
To get help with commands: helpcmds
To get help with utility information: helputils
Enjoy your stay!...

helputils---=Misc. Help=---
SCHizOS os a monolithic kernel based, very basic OS.
It contains some more advanced implementations like random,
tasks, processes, keyboard language changing, commands and some
other useful features.
I am open to suggestions to improvements.
```

Ez csak azt írja le, hogy nagyjából miket tartalmaz a rendszer. Ha viszont pontosabb információkra vágyik a végfelhasználó, akkor meghívhatja a „helpcmds” parancsot:

```
QEMU
Machine View
Enjoy your stay!...

helputils---=Misc. Help=---
SCHizOS os a monolithic kernel based, very basic OS.
It contains some more advanced implementations like random,
tasks, processes, keyboard language changing, commands and some
other useful features.
I am open to suggestions to improvements.

helpcmds---=Commands Help <page 1>=---
help/guide - Displays the main help text.
helpcommands -
helputils -
logout/shutdown/lo - shuts the OS down.
cls/clear - clears the screen.
lang <lang_id> - Changes keyboard language (EN is 0, DE is 1, HU is 2, FR is 3)
motd - displays the message of the day.
freemode/cmdmode - enables moving on the screen in all directions with the cursor.
random/rand/rnd - Gives you a random generated number.
hunglish - Generates you a new hungarian word.
dir/ls - Lists the files and directories in the current directory.
cat <file> - Prints a file content from a loaded file.
```

Ez már ténylegesen elmond információt a parancsok szintaxisáról és arról, hogy mit várnak paraméterül.

Ezen a ponton fontos megjegyezni, hogy a shell rendszer támogatja az alias-okat. Ez annyit jelent, hogy egy egzakt parancsot többféleképpen is meg lehet hívni. Például a „help” parancsot „guide”-kulcsszóval is meg lehet hívni:

```
guide---=Help=---
This command will guide through this tiny, sufferable adventure of yours.
Common command usage: <command> <arg1> ... <argN> (arguments are a must if they
are needed)
To get help with commands: helpcmds
To get help with utility information: helputils
Enjoy your stay!...
```

Az első tényleges parancs, ami nem csak információt mond el a végfelhasználónak, az a „logout” parancs. Ezt beírva az operációs rendszer „leáll”, ami azt jelenti, hogy már nem képes semmi új bemenetet kezelni, leállnak az event figyelők, a belső műkődések és a taszkok is leállnak:



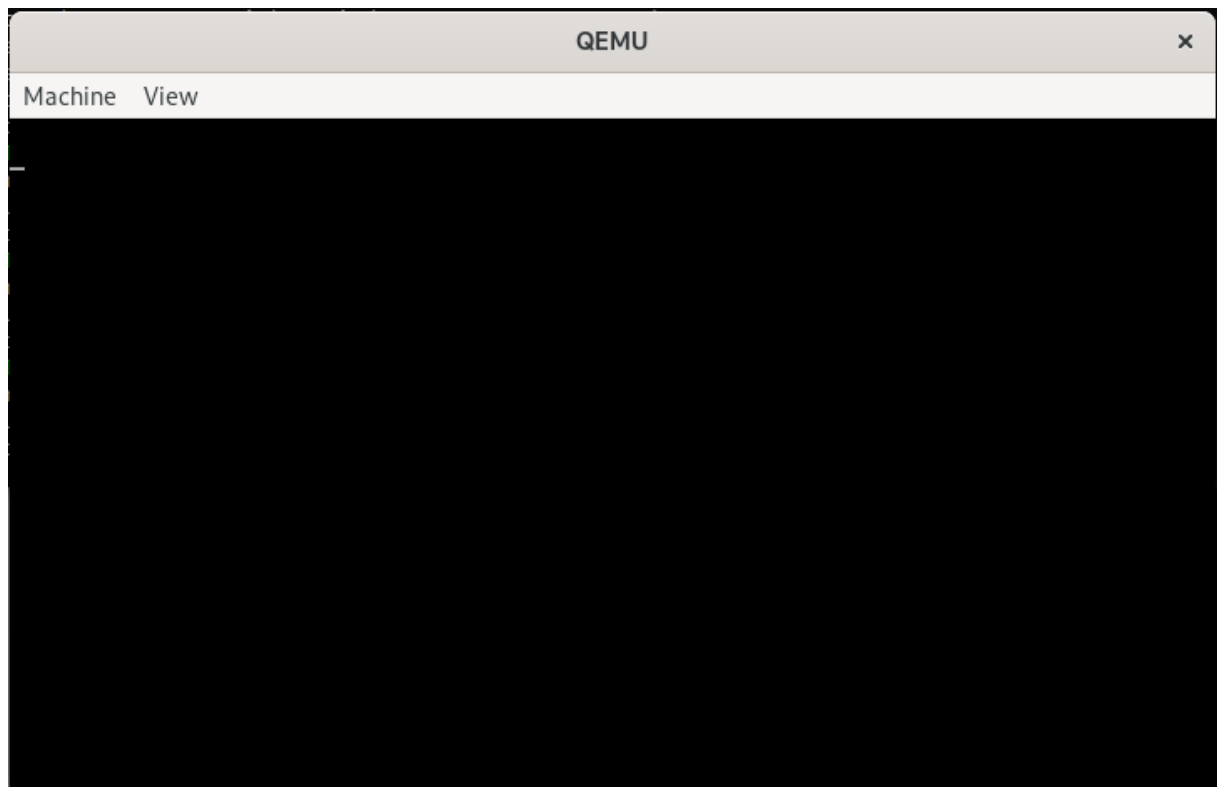
The screenshot shows a QEMU terminal window with a title bar that says "QEMU" and a close button "x". The terminal has a menu bar with "Machine" and "View". The main content area shows the output of the "helpcmds" command, which lists various system commands and their functions. The text is as follows:

```
Machine View
helpcmds---=Commands Help <page 1>=---
help/guide - Displays the main help text.
helpcommands -
helputils -
logout/shutdown/lo - shuts the OS down.
cls/clear - clears the screen.
lang <lang_id> - Changes keyboard language (EN is 0, DE is 1, HU is 2, FR is 3)
motd - displays the message of the day.
freemode/cmdmode - enables moving on the screen in all directions with the cursor.
random/rand/rnd - Gives you a random generated number.
hunglish - Generates you a new hungarian word.
dir/ls - Lists the files and directories in the current directory.
cat <file> - Prints a file content from a loaded file.

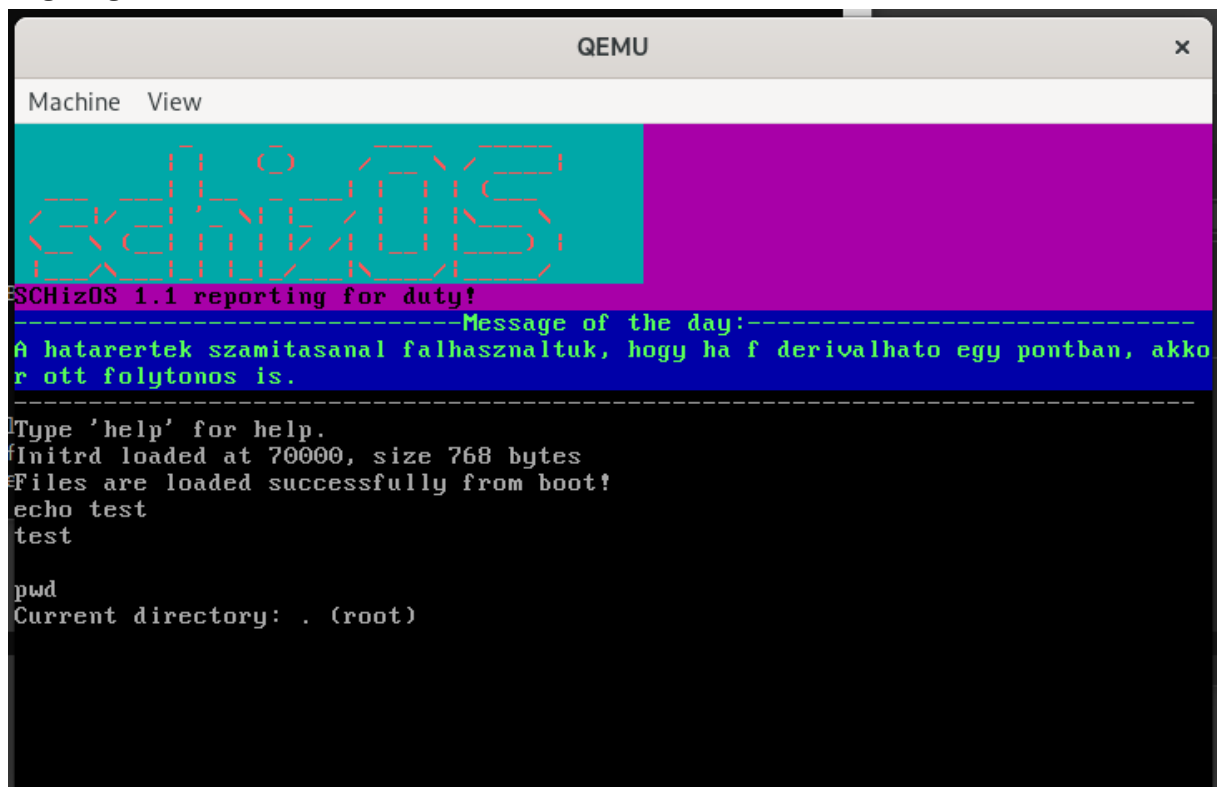
guide---=Help=---
This command will guide through this tiny, sufferable adventure of yours.
Common command usage: <command> <arg1> ... <argN> (arguments are a must if they
are needed)
To get help with commands: helpcmds
To get help with utility information: helputils
Enjoy your stay!...

logoutLogging out and shutting down...
```

Ezt a rendszer jelzi is a piros háttérű üzenettel. Ilyenkor csak bezárni tudjuk az emulátort, és ahhoz, hogy újra használni tudjuk a rendszert újra kell indítani az emulálást. A következő parancs a „cls”, ami letörli a képernyőről az összes karaktert és fekete háttérrel állít be:



Ezen felül az operációs rendszer képes a virtuális fájlrendszer navigálásra. Alapvetően a working directory a root könyvtárba van, erről meg lehet bizonyosodni a „pwd” parancs segítségével:





Kiegészítés: A jegyzőkönyv írása közben adtam hozzá a kibővített “cd”, “echo” és “pwd” parancsokat, ezeknek ez az újonnan leírása:

```
QEMU
Machine View

helpcmds---=Commands Help <page 1>---
help/guide - Displays the main help text.
helpcommands -
helputils -
logout/shutdown/lo - shuts the OS down.
cls/clear - clears the screen.
lang <lang_id> - Changes keyboard language (EN is 0, DE is 1, HU is 2, FR is 3)
motd - displays the message of the day.
freemode/cmdmode - enables moving on the screen in all directions with the cursor.
random/rand/rnd - Gives you a random generated number.
hunglish - Generates you a new hungarian word.
dir/ls - Lists the files and directories in the current directory.
cat <file> - Prints a file content from a loaded file.
cd <dir> - Enters in a directory.
pwd - Prints the working directory.
echo <message> - Prints a message.
```

Visszatérve, a navigálást segítő parancsok közül a “pwd” és a “dir” az egyik legszemléletesebb:

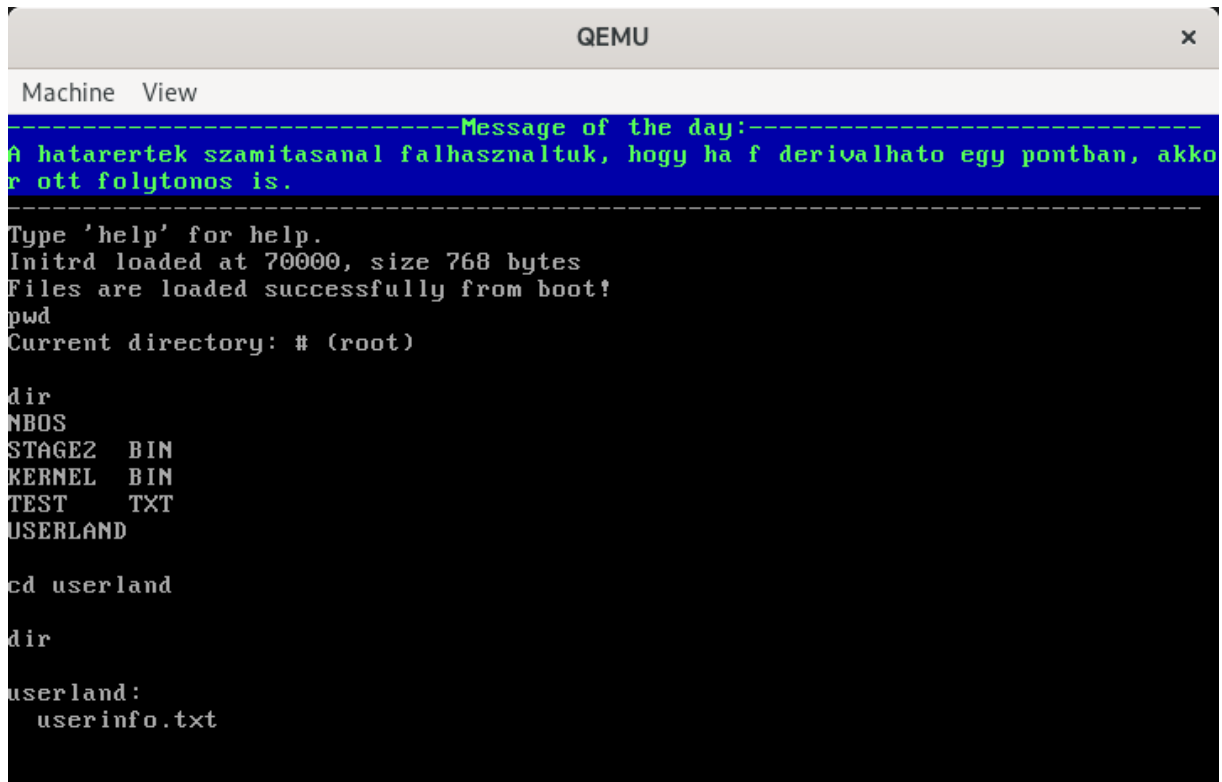
```
QEMU
Machine View

      _ _ _ _ _
     / / / / /
    / / / / /
   / / / / /
  / / / / /
 / / / / /
/ / / / /

SCHizOS 1.1 reporting for duty!
-----Message of the day:-----
A hatarertek szamitasanal falhasznaltuk, hogy ha f derivalhato egy pontban, akkor ott folytonos is.
-----
Type 'help' for help.
Initrd loaded at 70000, size 768 bytes
Files are loaded successfully from boot!
pwd
Current directory: # (root)

dir
NBOS
STAGE2  BIN
KERNEL  BIN
TEST    TXT
USERLAND
```

A „pwd” parancs jelzi, hogy a root könyvtárba van-e a felhasználó egy „(root)” szöveggel. A virtuális fájlrendszeremben a root könyvtár jele a „#”, Ami nem szabványos, de amikor azt készítettem az tűnt egy jó egyedi ötletnek, mivel ritkán van szükség '#' szimbólumra a fájlnevekben. A „cd” paranccsal be lehet lépni más könyvtárakba:



```
QEMU x
Machine View
-----Message of the day:-----
A hatarertek szamitasanal falhasznaltuk, hogy ha f derivalhato egy pontban, akkor ott folytonos is.
-----
Type 'help' for help.
Initrd loaded at 70000, size 768 bytes
Files are loaded successfully from boot!
pwd
Current directory: # (root)

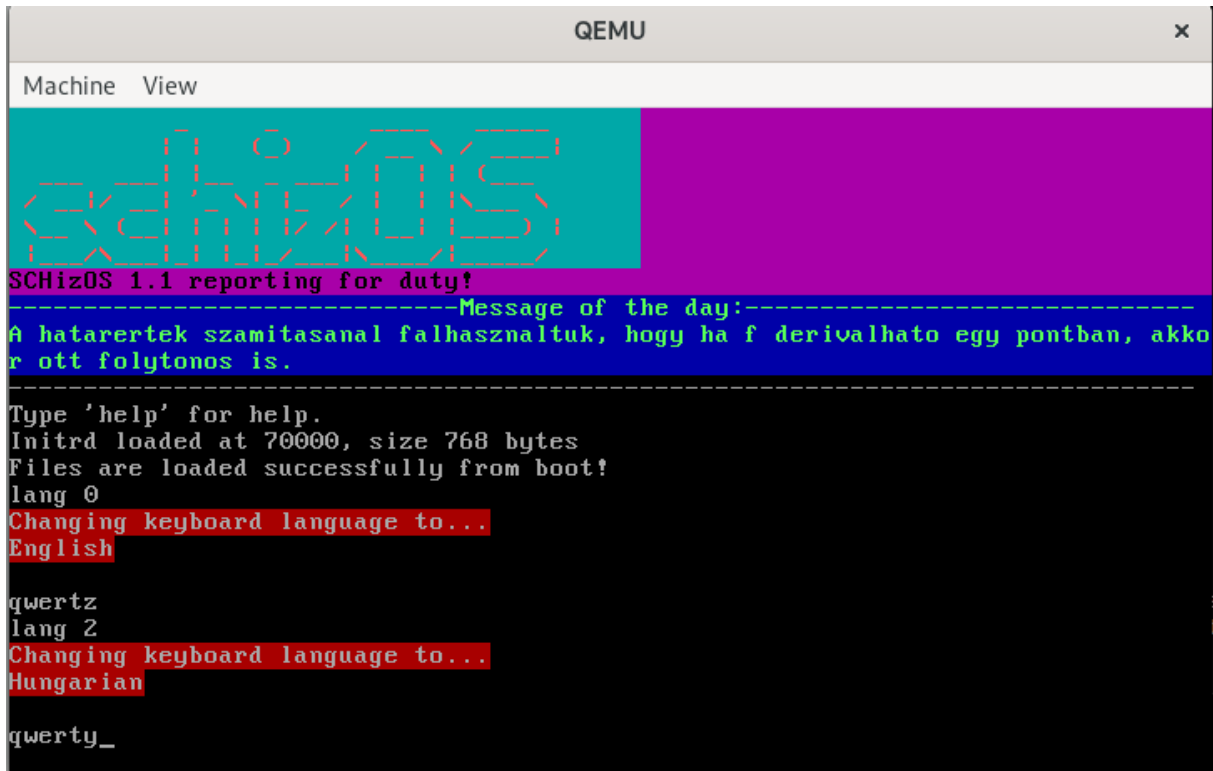
dir
NBOS
STAGE2  BIN
KERNEL  BIN
TEST    TXT
USERLAND

cd userland

dir
userland:
  userinfo.txt
```

A „dir” kicsit másképpen viselkedik, ha a root-ba van vagy máshol, ugyanis a rootba csupa nagybetűvel írja ki a fájlok nevét jelzésként, míg másik könyvtárakba „normálisan” (<fájlnév>.<kiterjesztés>) írja ki.

Az utolsó fontosabb parancs a „lang” parancs, ahol a billentyűzet nyelvét lehet állítani:



```
QEMU
Machine View
SCHIzOS 1.1 reporting for duty!
-----Message of the day:-----
A hatarertek szamitasanal falhasznaltuk, hogy ha f derivalhato egy pontban, akko
r ott folytonos is.
-----
Type 'help' for help.
Initrd loaded at 70000, size 768 bytes
Files are loaded successfully from boot!
lang 0
Changing keyboard language to...
English
qwerty
lang 2
Changing keyboard language to...
Hungarian
qwerty_
```

Itt látszik, hogy a “qwerty” gombok lenyomása más eredményt ad.

## Demo futás

Az operációs rendszer működéséhez számos demó scriptet is elkészítettem, melyeknek a célja, hogy az operációs rendszer bizonyos működését bemutassák. Ezek a scriptek a „demos” almappába érhetőek el, és az emberi viselkedést szimulálják (pontosabban, azt szimulálják, hogy az ember mely billentyűket üt le a gépen).

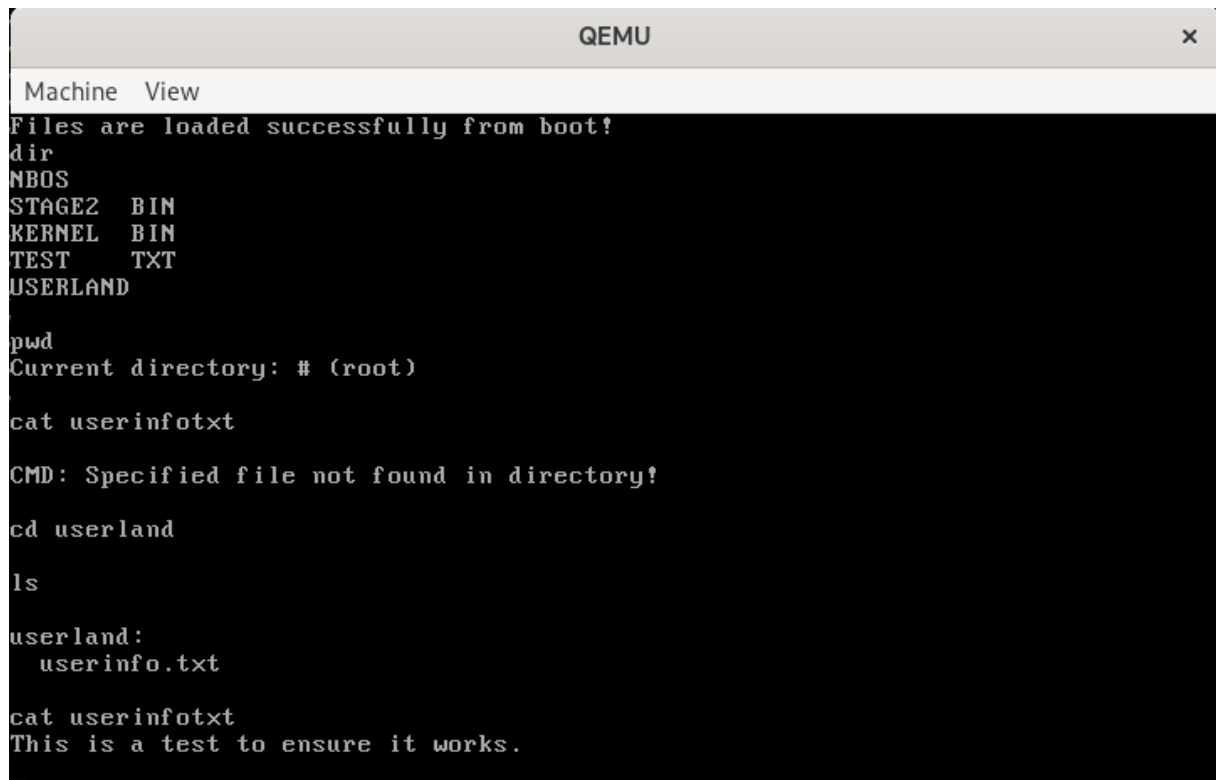
### 1. Demo: A felhasználó megkeres egy fájlt, és ki akarja iratni a tartalmát.

Ezt a demo-t a „demos” mappából így lehet elindítani:

```
./run_demo1.sh
```

A demo lényege az, hogy a „gép” egy keresést szimulál, ahol megnézi azt, hogy hol van ő most. A keresett fájl az „userinfo.txt” és ezt eleinte nem találja a root-ba. Ezek után átvált az „userland” mappába, amibe a fájl van és ott már kiíródik a fájl tartalma rendesen.

Ez a demo ilyesmi kimenetet ad:



```
QEMU
Machine  View
Files are loaded successfully from boot!
dir
NBOS
STAGE2  BIN
KERNEL  BIN
TEST    TXT
USERLAND

pwd
Current directory: # (root)

cat userinfotxt

CMD: Specified file not found in directory!

cd userland

ls

userland:
  userinfo.txt

cat userinfotxt
This is a test to ensure it works.
_
```

A demó fő lényege, hogy lássuk a virtuális fájlrendszer helyességét.

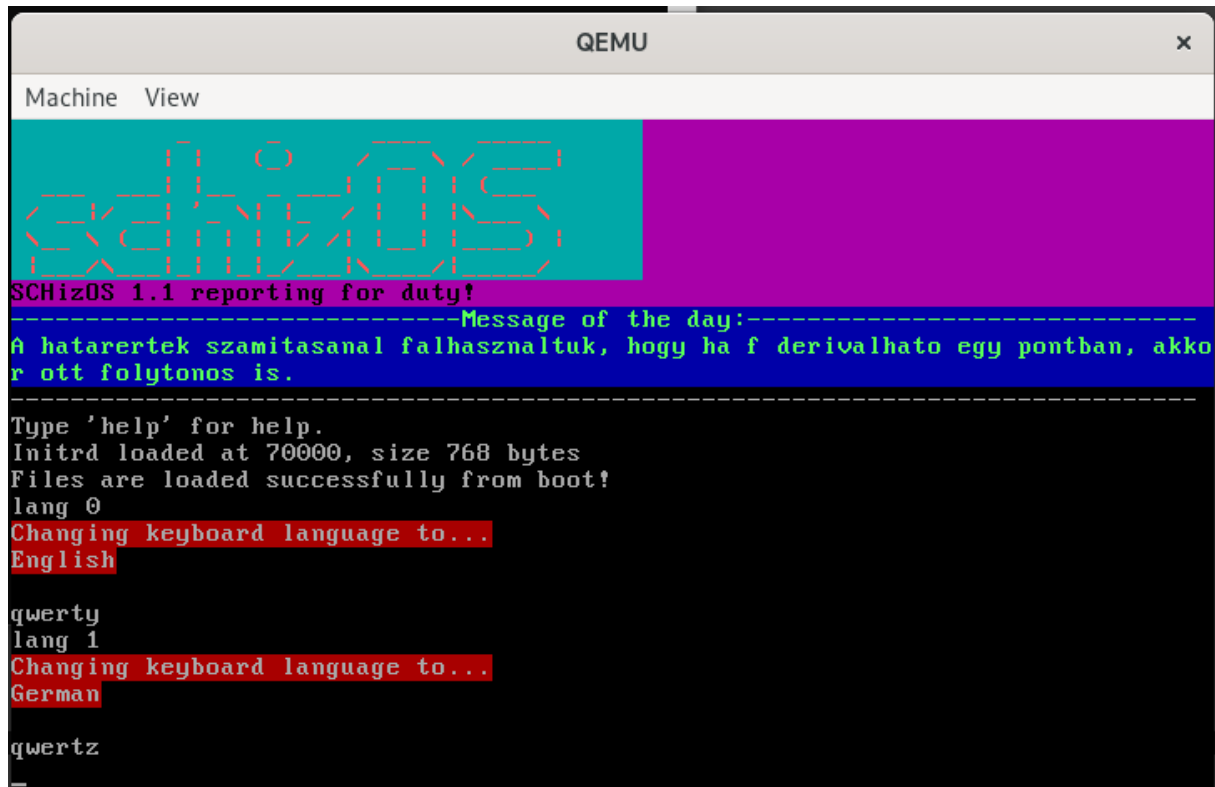
## 2. Demo: A billentyűzet választás tesztelése

Ezt a demo-t is hasonlóan a „demos” mappából így lehet elindítani:

```
./run_demo2.sh
```

Ennek a demo-nak a lényege az, hogy kipróbálja a német-angol nyelv váltást. Ez abban fog megtestesülni, hogy a „qwerty” gombok lenyomása más-más eredményt fog kiadni, mert a 2 billentyűzet nyelvezeten a z és az y fel van cserélve. A demo az alábbi kimenetet

adja:



## Taszkok kezelése

Az operációs rendszer egy relatívan buta taszk ütemezővel rendelkezik. A taszk ütemező minden futó taszkot futtat, ha az ütemezőnek megmondjuk, hogy „frissítsen”. Amelyik taszk a futása után „hamis” értéket ad vissza, azt az ütemező kiszedi a futó taszkok közül. Egy taszk úgy van leképezve, hogy egy igaz-hamis értéket ad vissza attól függően, hogy még a tick után aktív-e.

Fontos, hogy ez a szimpla taszk ütemező nem időosztásos, az ütemező kooperatív működésű: a taszkok önként adják vissza a vezérlést, így nincs időosztás vagy megszakítás-alapú váltás.

## Memóriakezelés menete

A rendszer-ben még nincs rendes virtuális memóriakezelés (van benne egy virtuális garbage collector alapú memória kezelő de az nagyon ritkán működik), csak 1/1 fizikai memóriacímzés van.

## A bootolás menete

A bootolásnak 2 fázisa van.

### 1. Fázis:

Előkészíti a 2. fázis betöltését és megvalósít egy nagyon barebones diszk olvasó függvényt. Ha megtalálta a 2. fázis bináris-át, akkor elkezdi betölteni a FAT-et (File

Allocation Table-t) és onnantól megkezdődik az úgynevezett Fájl Allokációs Tábla lánc feldolgozása. (A kernel olvasása is itt kezdődik).

Itt egy főbb hibát kezeltem le:

- floppy\_error: Ha a regiszterek mentése vagy a diszkel való kommunikáció sikertelen volt.

A lekezelés annyit jelent, hogy kiír a program egy „Hibaüzenetet”.

Fontos, hogy itt nagyon kevés helye van a programnak. Az MBR (Master Boot Record) csak 512 bájt lehet.

## 2. Fázis:

Itt történik a 32-bites protected módba való átváltás. Az alábbi ábra a betöltésért felelős assembly kódot mutatja:

```

global entry

;stack initialization
entry:
    ;PROTECTED MODE REQUIREMENTS:
    ;1. Disable interrupts and NMI interrupts (hardware failure, watchdog error)

    cli
    mov [global_boot_drive],dl

    ;init stack
    mov ax,ds
    mov ss,ax
    mov sp,0xFFF0
    mov bp,sp

    ;2. Enabled A20 gate
    call EnableA20

    ;3. Load Global Descriptor Table
    call LoadGDT

    ;4. Set enable protected mode flag in CR0
    mov eax,cr0
    or al,1
    mov cr0,eax

    ;5. Perform a far jump into protected mode
    jmp dword 08h:.pmode

.pmode:
    [bits 32]
    ;The is protected mode (32-bits)

    ;Changed:
    ;- All values in stack must be 32 bits wide
    ;- No segmented offset scheme
    ;- Flat memory mode
    ;- Functions params and return values must be 32 bits wide
    ;- Interrupts cause triple fold

    ;6. setup segment registers
    mov ax,0x10
    mov ds,ax
    mov ss,ax

    ;text can be drawn (at 16 colors) VGA interface
    ;clear BSS (uninitialized data so we initialize it)
    mov edi,__bss_start
    mov ecx,__end
    sub ecx,edi
    mov al,0
    cld
    rep stosb ;store string byte, repeatedly set things to 0

    ;expect boot drive in dl
    xor edx,edx
    mov dl, [global_boot_drive]
    push edx
    call start

    cli
    ;mov bl,1
    hlt

```

A protected módba váltás azt a főbb dolgot csinálta, hogy az értékek innentől 32 bit



hosszúak. (Eddíg 16 bit volt a real mód miatt).

Ezen felül a kernel végleges betöltése itt valósul meg. A rendszerindításhoz szükséges paramétereket is itt állítja be az OS.

Itt kéri le az OS a memória mennyiségét és egyéb adatait, amivel tud majd utána dolgozni.

## Végszó

Ez az operációs rendszer korán sem annyira életképes, mint egy Linux vagy akár egy másik enterprise rendszer. A használatát erősen ellenzem, ez a rendszer inkább csak egy orvosi ló, amin látszik, hogy egy minimális operációs rendszernek hogyan kell kinéznie. Fontosnak tartom, hogy a jövőben saját grafikus drivert készítsék a rendszerhez, habár ez egy nagyon hosszú folyamatot fog igénybe venni.