

## Régi idők UNIX-a

A SIMH rendszer segítségével megtapasztalhatjuk, hogyan működött a UNIX a digitális (h)őskorban.

Sokak szerint sokkal hajlamosabb vagyok a nosztalgiára, mint ami az életkoromból következne. Azt hiszem igazuk lehet. Bár 1976-ban születtem, rajongásig szeretem a korabeli nagygépek és miniszámítógépek történelmét tanulmányozni. Ami azt illeti, azt hiszem az emberiség egyik legnagyobb találmánya maga a történetírás. Úgy tűnik az emberi gondolkodásba bele van drótozva az a törekvés, hogy szeretjük följegyezni és elemezni elődeink cselekedeteit, és szeretünk tanulni a hibáikból és a sikereikből egyaránt. A nyílt forrás nem egyéb, mint ennek az elvnek az érvényesülése a számítástechnikában, hiszen ez a megközelítés lehetővé teszi, hogy tanuljunk mindazoknak a programozóknak a forráskódjaiból, akik előttünk éltek és alkottak egy-egy területen. Ha elemezzük egy korábbi fejlesztési projekt fejlődését, sokat tanulhatunk annak hibáiból, de ami még ennél is fontosabb, jó esélyünk van rá, hogy egyszerűen lemásolhatjuk belőle a sikeres részeket. Azzal pedig, hogy összeszedjük ezt a szabadon hozzáférhető történelmet, és profitálunk belőle, magunk részeseivé válunk a közös továbbfejlesztésének.

Számos nagyvállalat mostanában elkezdte nyilvánosságra hozni a tulajdonában levő „intellektuális javakat”. Bár egyesek nem teljesen tették magukévá a nyílt forrás filozófiáját, ezekből a gesztusértékű cselekedetekből mindenképpen sokat tanulhatunk. Ebben a cikkben azt szeretném bemutatni, miként tanulmányozhatjuk az operációs rendszerek történelmét azáltal, hogy saját *Linux* rendszerünkön ma már „történelminek” számító *UNIX* változatokat emulálunk. Jelenleg

a *SCO* csoport (bizony ők, korábban pedig a *Caldera Inc.*) tudja magáénak a korai *UNIX* változatok szerzői jogait, ám nem kereskedelmi célra szabaddá tette azokat egy úgynevezett „Ősi *UNIX* licenzmegállapodás” (*Ancient UNIX License*) keretében. Itt és most én a *UNIX V5* kiadására fogok koncentrálni, mivel a jelenleg hozzáférhetőek közül ez a legkorábbi. A *UNIX V6*, *V7*, valamint a *BSD* különböző korai kiadásai szintén rendelkezésünkre állnak. Mindazonáltal aki azt tervezi, hogy kipróbálja ezek közül valamelyiket, az a biztonság kedvéért azért olvassa el a licenzmegállapodást, mielőtt először bebootolná a rendszert.

### Idegenként egy idegen helyen: a *UNIX V5* felhasználói környezet

A lemezlenyomat (*disk image*) formájában rendelkezésünkre álló *UNIX V5* meglehetősen ridegnek és barátságatlannak tűnik a mai kényelmes *UNIX/Linux* rendszerekhez képest. Éppen ezért nem árt pár dologgal tisztában lenni, mielőtt belekezdénk az ismerkedésbe:

- A héjat itt még *sh*-nak hívják. Összesen 858 sornyi C kód az egész, tehát ne várjunk tőle semmi olyasmit, amit a *Bash*-tól megszoktunk!
- Ha könyvtárat akarunk váltani, használjuk a *chdir* parancsot.
- A *Backspace* és a nyílak a legritkább esetben működnek normálisan.
- A szövegszerkesztőt *ed*-nek hívják. Aki még nem hallott róla, az látogasson el a [en.wikipedia.org/wiki/Ed](http://en.wikipedia.org/wiki/Ed) címre.

- Rendelkezésünkre áll egy *Basic* interpreter, amit *bas*-nak hívnak.
- Hasonlóan van egy *Fortran* értelmező *fc* néven.
- A C fordító neve *cc*.
- A */usr/source* könyvtárban megtaláljuk a teljes rendszer forráskódját.
- Ami azt illeti nincs túl sok fájl az egész rendszerben, így a lista tanulmányozásához nyugodtan használhatjuk a *find / -print* parancsot.

Ahhoz, hogy tanulmányozhassuk az ilyen és ehhez hasonló operációs rendszereket, nyilván képesnek kell lennünk őket valamilyen módon futtatni a jelenleg használatos gépeken. Szerencsére vannak erre a célra megfelelő szimulátorok. Minősége és támogatottsága miatt ezek közül az egyik legnépszerűbb a *SIMH*, amit a <http://simh.trailing-edge.com> webhelyről tölthetünk le. A *SIMH* gyakorlatilag valamennyi manapság népszerű \*nix rendszeren, sőt még *Microsoft Windows* alatt is futtatható. Ami képességeit illeti, számos korai géptípust tud szimulálni, beleértve a *Digital Equipment Corp. PDP* és *VAX* rendszereit, az *MIT's Altair* nevű gépét, a korai *IBM* rendszereket, és még számos más öreg masinát. Ezek közül a történetileg legérdekesebbek talán a *DEC PDP* gépei, hiszen a *UNIX* ezeken kezdte meg pályafutását. A *SIMH* az alapoktól kezdve építkezik, vagyis képes szimulálni a korabeli gépek processzorát, memóriáját, beégetett programját (*firmware*) és minden csatlakoztatott

## 1. kód

```

$ mkdir simh
$ cd simh
$ unzip /path/to/simhv34-0.zip
$ mkdir BIN # Note all CAPS
$ gmake USE_NETWORK=1 all
# Only include USE_NETWORK=1 if your PCAP lib is up to date.
(compilation chatter omitted)
$ ls -l ./BIN/
total 11624
-rwxrwxr-x 1 matt matt 301959 Jul 16 18:45 altair
-rwxrwxr-x 1 matt matt 482274 Jul 16 18:45 altairz80
-rwxrwxr-x 1 matt matt 529317 Jul 16 18:44 eclipse
-rwxrwxr-x 1 matt matt 297590 Jul 16 18:45 gri
-rwxrwxr-x 1 matt matt 375737 Jul 16 18:44 h316
-rwxrwxr-x 1 matt matt 577678 Jul 16 18:44 hp2100
-rwxrwxr-x 1 matt matt 355225 Jul 16 18:44 i1401
-rwxrwxr-x 1 matt matt 381672 Jul 16 18:45 i1620
-rwxrwxr-x 1 matt matt 441079 Jul 16 18:46 ibm1130
-rwxrwxr-x 1 matt matt 502037 Jul 16 18:46 id16
-rwxrwxr-x 1 matt matt 508378 Jul 16 18:46 id32
-rwxrwxr-x 1 matt matt 294614 Jul 16 18:46 lgp
-rwxrwxr-x 1 matt matt 434940 Jul 16 18:44 nova
-rwxrwxr-x 1 matt matt 345034 Jul 16 18:41 pdp1
-rwxrwxr-x 1 matt matt 752055 Jul 16 18:43 pdp10
-rwxrwxr-x 1 matt matt 1055376 Jul 16 18:43 pdp11
-rwxrwxr-x 1 matt matt 474153 Jul 16 18:42 pdp15
-rwxrwxr-x 1 matt matt 459203 Jul 16 18:41 pdp4
-rwxrwxr-x 1 matt matt 460363 Jul 16 18:41 pdp7
-rwxrwxr-x 1 matt matt 499473 Jul 16 18:42 pdp8
-rwxrwxr-x 1 matt matt 467662 Jul 16 18:42 pdp9
-rwxrwxr-x 1 matt matt 352233 Jul 16 18:45 s3
-rwxrwxr-x 1 matt matt 429312 Jul 16 18:46 sds
-rwxrwxr-x 1 matt matt 982694 Jul 16 18:43 vax

```

eszközét. A *SIMH* tökéletesen utánozza a lemezegységeket, szalagos meghajtókat, sornyomtatókat és a hálózati eszközöket is. Ez utóbbi pedig ezt jelenti, hogy nem csak futtatni tudjuk ezeket az ősi operációs rendszereket, hanem adatokat is cserélhetünk segítségükkel a modern protokollokra támaszkodva. Mindezért a legnagyobb köszönet természetesen azokat illeti, akik hozzájárultak a *SIMH* fejlesztéséhez. Az ő munkájuk, no meg az az elhatározásuk, hogy annak gyümölcsét nyílttá teszik egyrészt hozzájárul ahhoz, hogy megérthessük saját digitális történelmünket, másrészt biztosítja, hogy ez a történelem mindörökre hozzáférhető maradjon valamennyiünk számára.

### A *SIMH* telepítése

Kezdésként töltsük le a *SIMH* legfrissebb változatát (ez jelenleg a **3.4-0**-ás verzió), fordítsuk le és telepítsük. Aki az *Ethernet* emulációt is használni szeretné, annak előbb valószínűleg frissítenie kell majd a saját operációs rendszeréhez kapott *libpcap* könyvtárat, mivel ez a legtöbb jelenlegi terjesztésben meglehetősen elavult. Ennek a műveletnek a leírását részletesen tartalmazza a *SIMH* dokumentációja, ha azonban nincs szükségünk a kérdéses szolgáltatásra, akár ki is hagyhatjuk ezt a lépés. A fordítást egyszerű felhasználóként is elvégezhettük a következő parancsok segítségével (1. kód). Ezzel az összes jelenleg támogatott rendszer szimulátorát lefordítottuk.

Mint látható, mindegyik egy-egy külön bináris állományt jelent a *./BIN* könyvtárban. Magát ha *SIMH* rendszert bármely felhasználó futtathatja, ha azonban *Ethernet* emulációt használunk, akkor muszáj rendszergazdaként futtatni, máskülönben a *libpcap* nem fér hozzá a valódi *Ethernet* eszközöz.

### A *UNIX V5* futtatása

Az 1974 júniusában kiadott *UNIX V5* még meglehetősen korai műve volt a *Bell Labs*-nak. Ezt jelzi az is, hogy a rendszer java részét még assemblyben írták. Ez a lemezkép ugyanakkor már tartalmaz egy *C* fordítót (*cc*) a */usr/source* könyvtárban pedig rengeteg érdekes forráskódot találhatunk. A felfedezés megkezdéséhez előbb le kell töltenünk a *UNIX V5* lemezképét (lásd a forrásokat). Ez egy zip tömörítvény, ami tartalmazza az előtelepített rendszert és egy *README* fájlt, amiben a szokásos licenc információt találjuk. A lemezkép tulajdonképpen egy működő rendszerről készült pillanatkép. Kicsit konkrétan rendszerünk egy *RK05* lemezegységet fog szimulálni. Először s össze kell gyűjtenünk azokat az elemeket, amelyek a rendszer bebootolásához szükségesek. Hozzunk létre egy új könyvtárat, majd másoljuk bele a *BIN/pdp11* bináris fájlt, valamint az *uv5swre.zip* fájl tartalmát. Aztán egy tetszőleges szövegszerkesztővel hozzuk létre a *pdp11.ini* fájlt, ami a szimulátor működését fogja majd vezérelni. Az *ini* fájlnak a következőket kell tartalmaznia:

```

set cpu u18
attach rk0 unix_v5_rk.dsk
boot rk0

```

Ez a beállítófájl mondja meg a szimulátornak, hogy milyen típusú processzort emuláljon, illetve hogy esetünkben csatlakoztassa a *unix\_v5\_rk.dsk* fájlt egy szimulált *RK*-rendszerű lemezegységként, az ehhez tartozó eszköz neve pedig legyen rk0. Az utolsó sor szerint magát az operációs rendszert is erről a lemezzel kell bebootolni. A kérdéses könyvtárnak tehát a végén így kell kinéznie (2. kód).

## 2. kód

```
-rw-rw-r-- 1 matt matt 12299 Jan 24 2002 AncientUnix.pdf
-rwxrwxr-x 1 matt matt 913614 Jul 22 19:33 pdp11
-rw-rw-r-- 1 matt matt 47 Jul 22 23:59 pdp11.ini
-rw-rw-r-- 1 matt matt 263 Nov 25 1996 README.txt
-rw-rw-r-- 1 matt matt 2494464 Jul 23 00:39 unix_v5_rk.dsk
```

## 3. kód

```
$ ./pdp11
PDP-11 simulator V3.4-0
Disabling XQ
@unix
login: root
# ls -l /
total 60
drwxr-xr-x 2 bin 944 Nov 26 18:13 bin
drwxr-xr-x 2 bin 80 Nov 26 18:13 dev
drwxr-xr-x 2 bin 240 Mar 21 12:07 etc
drwxr-xr-x 2 bin 224 Nov 26 18:13 lib
drwxr-xr-x 2 bin 32 Nov 26 18:13 mnt
drwxrwxrwx 2 bin 112 Mar 21 12:11 tmp
-rwxrwxrwx 1 bin 25802 Mar 21 12:07 unix
drwxr-xr-x 14 bin 224 Nov 26 18:13 usr
# chdir /usr/source/sl
# cat echo.c
main(argc, argv)
int argc;
char *argv[];
{
    int i;
    argc--;
    for(i=1; i<=argc; i++)
        printf("%s%c", argv[i], i==argc? '\n': ' ');
}
# cc echo.c
# mv a.out newecho
# ./newecho Hello world
Hello world
# chdir /tmp
# cat >hello.c
main()
{
    printf ("Hello world!\n");
}
# cc hello.c
# ./a.out
Hello world!
# cat >hello.b
10 print "Hello world!"
# bas hello.b
run
Hello world!
```

A **UNIX V5** rendszer bebootolásához most már csak a `./pdp11` parancsot kell kiadnunk. Aztán amikor megjelenik a `@` prompt, gépeljük be a `unix` parancsot. Erre gyakorlatilag azonnal megkapjuk a bejelentkezési promptot, a korabeli **UNIX** ugyanis bootolás közben egyáltalán nem voltak olyan beszélgetések, mint a mostaniak. Szimulált rendszerünkön root jelszó nincs, így azonnal egy parancspromptot kapunk. Az egész folyamat a következőképpen néz ki a képernyőn (3. kód). Ennyi kérem! A rendszerünk máris működőképes így elkezdhetünk tanulmányozni egy „igazi” történelmi **UNIX** operációs rendszert. Amint látható rengeteg érdekes forráskódot találunk, és van hozzá egy **C** fordítónk is, hogy játszhassunk vele. A **UNIX V5** csupán az egyik azok közül a régi rendszerek közül, amelyeket a **SIMH** szimulátoron keresztül megismerhetünk. A **SIMH** hivatalos webhelyén lemezképek egész gyűjteményét találjuk, amelyek mind egy-egy ilyen régi rendszert tartalmaznak. Aki pedig látni szeretné, hogyan is nézett ki a valóságban egy **PDP-11** vagy egy **RK05** lemezegység, vessen egy pillantást a fotogalériára. A **Google** képkeresőjével szintén számos e témával kapcsolatos érdekes fotót találhatunk.

*Linux Journal 2005. 140. szám*



**Matthew Hoskins**

vezető UNIX rendszergazda a New Jersey-ben található Institute of Technology-ban, ahol ő felelős a legtöbb adminisztratív rendszerért. Élvezi, amikor egészen vadul eltérő rendszereket és programokat hangolhat össze, és bírhat együttműködésre. Ehhez rendszerint egy Perl-ben megírt vékony összekötő réteget használ, amit a helyiket MattGlue (Matt ragasztója) néven emlegetnek. Amikor nem a különböző rendszereket hackeli, általában a konyhában foglalatосkodik hasonlóan érdekfeszítő dolgokkal. Matt a `matt@njit.edu` címen érhető el.

## KAPCSOLÓDÓ CÍMEK

➔ [www.linuxjournal.com/article/8587](http://www.linuxjournal.com/article/8587)