Linux a tanteremben – 3. rész

Amennyiben a jövő szempontjából értékesnek tartjuk a Linuxot, a jövő nemzedékét kell felkarolnunk, hogy elsajátítsák a Unix-látásmódot, és megtanulják kezelni a Unix-rendszereket.

A sorozat zárásaként a szegedi Radnóti Miklós Gimnáziumban az elmúlt tanévekben tartott Linux-szakkörökről szóljon egy rövid beszámoló.
„Nálunk több szakkörpróbálkozás is volt, és mindegyik dugájába dőlt" – olvastam nemrég a SuLinux levelezési listán egy neves linuxos rendszergazda-tanár sorait. Nálunk a tanév eleji lelkesedés sokszor nagyobb volt, mint a későbbi részvétel: a gyerekek között ugyanúgy „divat" a Linux használata, ahogyan az különc módon öltözködni; azok, akik azonban komolyan is gondolják, hogy hátat fordítanak a Microsoft-világnak, valójában kevesen vannak. Szerencsére már nem mindenütt, az idézett rendszergazda-tanár középiskolájában a nyári szünet elején tíz nap alatt száznál is több különböző felhasználó jelentkezett be a Linux-kiszolgálóra. Nálunk ez a szám a teljes nyári szünet alatt öt alatt volt…
A Linux-szakkört azért többen látogatták, csakhogy szinte mindig mások. A gyerekek tudásszintje egymáshoz képest lényegesen különböző volt, így lehetetlen volt olyan órát tartani, amiből mindegyikük nyert volna valamit. Többször is telepítettük a Linuxot (először még abban az időben, amikor a 6.1-es SuSE kijött), tanítottam könnyebb héjprogramozást a *dialog* program bemutatásával, egyszerűbb programokat készítettünk együtt Kylixban és PHP-ben (PostgreSQL-támogatással), (a legtöbb anyagot az ➲ http://ftp://pc10.radnoti-szeged.sulinet.hu/home/kovacsz/Linux címen nyilvánossá tettem). A gyerekek számára mindez nem volt kivételesen nagy élmény, azonban remélem, néhányójuknak induló lökés is lehetett. A linuxos tudást a gyerekek azonban nem a szakkörön gyűjtik össze, hanem saját tapasztalataik alapján az otthoni számítógépük előtt.

A projektmunka és a Linux

Ezidáig a tehetséges gyerekek tanításának legösztönzőbb eszközét a közös játékprogram-fejlesztésben találtam meg. Még középiskolás voltam, amikor számítástechnika tanárom ugyanezzel a módszerrel indított el bennünket a véresen komolyan vett programfejlesztés felé, abból a célból, hogy hatékonyan készüljünk fel az Országos Középiskolai Tanulmányi Versenyre. Az egyetemen is volt olyan gyakorlatvezetőm, aki programozás tárgyból kötelezően beadandó programként logikai játék

```
#include "allegro.h"

BITMAP *kocsi, *hatter;
 /* Az aut  Øs a hÆttØr. */
PALLETE paletta; /* Az Øppen ØrvØnyes
➥sz npaletta. 255 sz nß, a 0. Ætlætsz  */


void mozgat()
 {

 BITMAP *megjelenitendo; /* Ebben tÆroljuk
➥a kØpernyı vÆltoztatand  rØszØt. */

 char c; /* A megnyomott gomb. */
 int lepeskoz=1; /* Az aut  mozgÆsÆnak
       ➥gyorsasÆga. GØptıl is f gg. */

 int kepatlo,irany=192;
 /* Aut  "ÆtmØrıje", indul  irÆny (Øszak). */
 float x,y,iranyx=-1,iranyy=0;
 /* Aut  koordinÆtÆi, indul  irÆnya. */
 kepatlo=sqrt((kocsi->w)*(kocsi->w)+
           ➥(kocsi->h)*(kocsi->h));
 // Pitagorasz-tØtel!
 kepatlo+=5; /* Egy kicsit megn velj k, hogy
            ➥fordulÆsnÆl is szØp legyen. */

megjelenitendo=create_bitmap(kepatlo+lepeskoz*2,
➥kepatlo+lepeskoz*2);

 /* AzØrt adjuk mØg hozzÆ a lØpØsk zt, hogy
    ➥nagy gyorsasÆgnÆl ne fussunk ki. */

 x=(320-(megjelenitendo->w))/2;
 /* BeÆll tjuk az aut  indulÆsi poz ci jÆt. */
 y=(200-(megjelenitendo->h))-40;

 do
 {

  if (key[KEY_LEFT]) irany-=2; /* balra */
  if (key[KEY_RIGHT]) irany+=2; /* jobbra */
  if (key[KEY_UP])
 {
 /* elıre */
   y-=((float)lepeskoz*iranyy);
   x-=((float)lepeskoz*iranyx);

   }

  if (key[KEY_DOWN])
 {
 /* hÆtra */
   y+=((float)lepeskoz*iranyy);
   x+=((float)lepeskoz*iranyx);

   }

iranyx=cos((float)((((float)irany+64)/128)*3.1415));
 /* koszinusz... :-) */
  iranyy=sin((float)((((float)irany+64)/128)*
➥3.1415));
 /* szinusz... :-) */
  /* Feltessz k a hÆttØr megfelelı darabjÆt
  ➥a vÆltoztatand  rØszre: */

blit(hatter,megjelenitendo,(int)x,(int)y,0,0,
➥kepatlo+lepeskoz*2,kepatlo+lepeskoz*2);
```

A lista folytatása a következő oldalon található

elkészítését írta elő. Érdekes, hogy ez az ötlet sokszor már a 13 éves gyerekeknél is bejön, habár logikai játék helyett eddig mindhárom alkalommal egy-egy akciójátékot kezdtünk el írni.
Az első próbálkozáson öt évvel ezelőttre tehető. Ekkor még csak próbaképpen tartottam szakkört a Radnótiban, negyedéves egyetemistaként. A szakkört a tanév közepén indítottam, és megbeszéltük a négy, nálam néhány évvel fiatalabb fiúval, hogy valami igen komoly stratégiai játékot írunk Turbo Pascalban, assembly-betétekkel. Néhány segédeljárás el is készült, összeállt némi grafikai terv is, de a program sohasem került futtatható állapotba.
A második próba három évvel ezelőtt volt, amikor a „A vágtázó halottkémek" fedőnevű projektben hét tanulóval egy ténylegesen működő, animált mozgást is tartalmazó, DOS és Linux alatt is futtatható kódot dobtunk össze, sok grafikával. A játék „Radnótis kaland" néven futott, és a lényege pedig az volt, hogy a főhőssel a suliban mászkálva kell mindenféle feladatot teljesíteni. Két szakkörös gyereket fényképeztünk le több fázisban a fehér fal előtt, majd a fényképeket beolvastuk, a Gimppel „megfésültük", s ezután a megfelelő C nyelvi utasítással a gimnázium szintén beolvasott jellemző részletei elé másoltuk őket (➲ //ftp://pc10.radnoti-szeged.hu/home/kovacsz/C_szakkor/JÁTÉK!).
Ez a második projekt tehát már C nyelven futott. A DOS-os DJGPP (➲ http://www.delorie.com/djgpp) alatt az Allegro-csomag (➲ http://allegro.cc) segítségével írtuk a programkódot, a szövegkiíró és a perspektív megjelenítésért felelős eljárást pedig egy-egy 14 éves tanítványom alkotta. Habár a programnak csak egy „nagyon példa" változata lett készen, a projektet mindenképp sikerként könyvelem el. Az Allegro csomagot sajnos kevesen ismerik; ez egy könnyen használható, gyors programfejlesztést lehetővé tévő eljárásgyűjtemény, amelynek fő erőssége a gyors grafikai megjelenítés és a több felületen való felhasználhatóság.
Annyira gyorsan lehet benne fejleszteni, hogy tavaly egy olyan projektet indítottam benne el, ami csak hetvensoros volt, mégis már így is működő autóversenyprogram lett. Ma is szívesen játszanak vele kicsik és nagyok egyaránt.
Ezt a programot kezdtük aztán Maxirace néven fejleszteni (a nevet egy hetedikes tanítványom adta) a SourceForge-on, CVS-támogatással. Ez utóbbihoz már Linux kellett; bár DOS alatt is lehetett volna CVS-t

A lista folytatása az előző oldalról

```
  /* A vÆltoztatand  rØszre rÆrajzoljuk az aut t: */

rotate_sprite(megjelenitendo,kocsi,lepeskoz+
➥(kepatlo-kocsi->w)/2, lepeskoz+(kepatlo-
➥kocsi->h)/2,itofix(irany));

  sync(); /* Ez a sor biztos tja az animÆci
         ➥villogÆsmentessØgØt. */

draw_sprite(screen,megjelenitendo,(int)x,(int)y);
 /* KIRAJZOL`S! */
  rest(5); /* 5 ezredmÆsodperc vÆrakozÆs. */
  }
 while (0!key[KEY_ESC]);
 /* KilØp nk, ha ESC volt a megnyomott gomb. */
 }


void inicializalas()
 {

 allegro_init();
 /* Ez minden grafikus program elejØre kell. */
 install_keyboard();
 /* Ez mindig kell, ha a billentyßzetet
    hasznÆljuk. */
 install_timer(); /* Ez mindig kell, ha
                     idız tØst hasznÆlunk. */
 hatter=load_bitmap("hatter.bmp",paletta);
 /* HÆttØrkØp beolvasÆsa. */

 kocsi = load_bitmap("auto.bmp", paletta);
  /* Kocsi beolvasÆsa. */

 set_gfx_mode( /* Grafikus m d beÆll tÆsa. */
#ifdef LINUX
  GFX_XWINDOWS,  /* Linuxon X Window alatt
                ➥fusson a program, */
#else
  GFX_AUTODETECT, /* egyØbkØnt legyen az
                ➥alapØrtelmezett m d. */
#endif
  320, 200, 0, 0);
 /* FelbontÆs. */
 set_pallete(paletta);
 /* Sz npaletta beÆll tÆsa. */
 draw_sprite(screen,hatter,0,0);
 /* Kirajzoljuk a hÆtteret. */
 }


main()
 {
 /* Fıprogram. */
 inicializalas();  /* Megh vjuk az
 ➥inicializalas() rØszprogramot. */
 mozgat();
 /* Elkezdj k a tØnyleges jÆtØkot. */
 }

#ifdef LINUX
END_OF_MAIN(); /* Ez Linuxon kell,
               ➥DOS alatt nem. */
#endif
```

telepíteni, a 7.3-as SuSE-ban minden konyhakészen rendelkezésre állt, az Allegróval együtt. A Maxirace program fejlesztésébe ezután sok apró fiatal kapcsolódott be: néhányan weboldalt készítenek hozzá (részint Windows alatt), mások háttereket és kocsikat rajzolnak, a legügyesebbek pedig a kódba is belenyúlnak. A CVS logikáját még nem sikerült megszerettetnem: környezeti beállításaink még nem elég kényelmesek ahhoz, hogy önműködően *update*-tel és *cvs commit*-tal tartsuk a kapcsolatot a CVS-kiszolgálóval (először be kell `ssh`-zni a *pc9*-re, csak onnan lehet elérni a SourceForge kiszolgálóját, majd a letöltés után a friss anyagot át kell másolni a grafikus munkaállomásra, végül a módosított fájlokat vissza kell tenni a *pc9*-re, s csak ezután indulhat a visszatöltés a raktárba). Ellenben rengeteg apróságot és szépséget el lehet mondani a gyerekeknek, mialatt a „nagy cél" elérése érdekében kis célokat tűzünk ki, s azokat oldjuk meg. A hetente tartott kétórás szakkör magját egy ötfős tevékeny csoport alkotja, hozzájuk verődik 2–3 főnyi „kíváncsiak társasága", azonban a hivatalos fejlesztői csapat nyolctagú: az egyik csapattag 19 éves, a többiek viszont mind 13–14 évesek. Legalább tíz fő a további „pártoló tagok" száma, ezen felül az iskola tanulóinak egy része véletlenül is elindítja olykor a Windows alatti változatot.

Allegro, azaz vidáman, gyorsan

A listán olvasható a 0.2-es változat forrása, ami igazából a legelső változat, csak megfésült, megjegyzésekkel ellátott formában. A jelenlegi, a ➲ http://sf.net/projects/maxirace címről letölthető ennél már ügyesebb, és a kód is profibb formájú. Már ez a programlista is jól mutatja azonban az Allegro egyszerűségét, és a

```
gcc -DLINUX -o maxirace.exe
➥maxirace.c 'allegro-config --libs'
```

parancssorral gond nélkül lefordul. A kimenet a *maxirace.exe* lesz, a bináris kódot a gyerekek miatt a futtathatóságot hangsúlyozandó *.exe* kiterjesztéssel hozzuk létre; az utolsó órákon a kiterjesztést már elhagytam. Az `allegro-config --version` parancssorral vizsgálhatjuk meg, hogy Allegro-csomagunk kellően friss-e a fordításhoz; nekem a program a 3.9.37-es változattal gond nélkül lefordult. A pillanatnyi könyvtárban helyezzünk el egy *hatter.bmp* nevű 320×200-as képet (lehetőleg kacskaringós versenypályával) és egy ennél jóval kisebb *auto.bmp*-t, ami jól elfér a versenypályánkon. Ezután a játék már indítható is: autónkat a nyílbillentyűkkel lehet irányítani az autóverseny-programokban megszokott módon, kilépni az Esc-pel tudunk. Programunk X-ablakban fut, de a `GFX_XWINDOWS` átírásával megoldható, hogy például Svgalib alatt fusson. Ha DOS alá kívánunk fordítani (DJGPP-vel, elegendő hozzá a 2.01-es változat), módja:

```
gcc -o maxirace.exe maxirace.c
➥-lalleg
```

A Maxirace kódja kellően rövid ahhoz, hogy ne rettentse el a kezdő programozókat, de elég hosszú és elég nyitott ahhoz is, hogy egy néhány hónapos programozói alapképzéshez ugródeszka legyen. Emellett nemcsak autóverseny-program kiinduló alapja lehet: a kétdimenziós játékok egy része egy tengely körüli forgásra és előre-hátramenetre épül, az alapprogram tehát számos irányba továbbfejleszthető.

Az idén januártól már nem is Linux-, hanem Maxirace-szakkört hirdettem a gyerekeknek. Nagyon jó hangulatú műhelymunkában volt részünk ezekkel a fiatalokkal hétről hétre, és ha lassan haladtunk is előre, az eredmény magáért beszélt.

E sorok írásakor, augusztus végén azt tervezem, hogy a programot ősztől kezdve továbbfejlesztjük. A SourceForge-os projektbe természetesen bárki bekapcsolódhat, aki ezek után kedvet kapott hozzá. Tanítványainkkal talán mégis a legjobb új projektet kezdeni, akár a SourceForge-on, akár saját CVS-kiszolgálón, akár CVS-kiszolgáló nélkül. Megítélésem szerint az öröm forrása a közös munkában a létrehozott termék varázsában van, abban, hogy mindenki megtalálhatja a hozzá illő, képességeinek megfelelő részfeladatot, de ha kíváncsi mások munkájára, az sincs elrejtve előle.

Zárszó

A nyílt forráskódra épülő rendszereknek számos előnyük van, de az oktatói-nevelői munka kapcsán egyet mindenképpen ki kell emelnem: a tájékoztatás, a tudás szabadságát. A zárt forrású kereskedelmi programok meghatározásukból adódóan megfosztják a felhasználót attól, hogy megtudja, „mitől is megy" a program. Ma már olyan bonyolult rendszerekre van szükség, hogy akármilyen ügyes is egy projekt (a legjobb szakembereket gyakran a Microsoft vásárolja fel), tökéletes programot alkotni képtelenség. Ez az adat visszatartásának zsákutcája. Ha egy tanár szakmailag felkészült szeretne maradni, félmegoldás, ha csak annyit mond: „nem tudom, miért nem működik" vagy „fogalmam sincs, mitől fagy le folyton". A nyílt rendszerek tanulmányozhatók, és a tanulás legmélyebb formáját, az önképzést segítik elő. Valljuk be, diákjaink némelyik területen sokkal többet tudnak a Linuxról, mint mi, a tanáraik. Sajnos sok tanár éppen a szakmai hírnevét félti, s nem közösködik a fiatal kölykökkel, akik „kenik-vágják a témát". De mindig öröm újabb és újabb pedagógusoktól hallani, hogy „igen, én félre mertem tenni, hogy nagynak hittem magamat". És ha tudományról, azaz számítástudományról beszélünk, az új ismeret mércéje mindenképpen a nyilvánosság. Ebből az előremutató, nyílt versenyből a zárt rendszerek, egyszersmind a bezárkózó tanárok is mind kimaradnak.

Kovács Zoltán
(kovzol@math.u-szeged.hu)
tanársegéd a Szegedi Tudományegyetem Bolyai Intézetében az Analízis Tanszéken, matematikát és számítástechnikát tanít óraadóként a szegedi Radnóti Miklós Kísérleti Gimnáziumban.