

A Linux mint játékplatform – múlt, jelen, jövő

A Linux története az első rendszermag megjelenésétől kezdve példaértékű. A fejlesztői közösség lelkesedésének hála, a nagy számú disztribúció közül némelyik ma már igen komoly alternatívát nyújt a mindenkori piacvezető termékek ellenében.

© Kiskapu Kft. Minden jog fenntartva

■ Az elmúlt években alkalmazott rendszermodell már számtalan helyen bizonyított, többek között az otthoni szegmensben is, ahol leginkább multimédiás és szórakozási céllal, esetleg kisebb irodai munka erejéig ülünk képernyőnk elé. Ebben a régióban érezhetően nyomja a képzeletbeli mérleget a játékok piaca is, hiszen sokak szerint „nem is igazi rendszer” az, mely alatt nem lehet egy kiadósat játszani. És valóban. Ugyan, melyikünk ne szeretné néha egy jó játékprogrammal agyonütni az időt? Játékok tekintetében azonban a *Linux* kétségtelenül elmarad a *Windows* mögött, ami persze nem jelenti azt, hogy nem lenne sok népszerű kereskedelmi játéknak linuxos változata is.

A közelmúlt

Nézzük röviden a „száraz” tényeket! A *Linuxot* kezdetben nem lehetett befogni otthoni használatra, vagy multimédiás platformnak, így játékprogramok alaprtegeként sem szolgált. A dolog mögött számos tényezőt bújít meg, kezdve a kezdetben kis létszámú felhasználói táborától egészen a megfelelő programozási felületek hiányáig. Mindez azért nem tekinthető igazán a negatívumnak, mert a *Linux eredetileg egyáltalán* nem szórakozási céllal született. Ennek megfelelően sokáig csak a „komolyabb” projektek profitáltak a stabilitásából és az áttekinthető szoftvermodellből. Volt persze néhány grafikus felületet nem igénylő „játékprogram” is,

de a „könnyed kikapcsolódás” lehetősége valójában a hatékony *X* kiszolgáló megjelenésével vált kézzel foghatóvá. Mint mindenütt, itt is akadt néhány a szabályokat erősítő kivétel, mint például a *Doom*. Az ezt fejlesztő *id Software* a játék *MS-DOS*-ra szánt változata (1993) után két évvel kiadott egy linuxos binárist is, a csapat pedig jelenleg is azok közé tartozik, akik komolyan gondolják a szabad szoftver ilyen támogatását.

Az út innen egyenesen vezetett a komolyabb műfajok felé, ami előtt a falat valójában azok a programozási felületek segítettek áttörni, melyek által a fejlesztők (már munkájuk kezdetén) komolyan fontolóra vehették a független kivitelezést és használhatóságot. Ahogyan telt az idő, az *SDL* keresztplatformos környezete, az *OpenAL* és *OpenGL* programozási felületek (sok más rendszerkomponenssel együtt) oly mértékben váltak használhatóvá, ami már hosszú ideje megkérdőjelezhetetlenül stabil és komoly alapot biztosított a fejlődésnek. Ezekre alapozva, a születendő játékprogramok *Linux* alatt közel ugyanazt a teljesítményt, látványvilágot és fizikát képesek elérni, melyet a *Windows* korszerű verziói alatt elérhetnek.

Kitaposott út

Linux felületen a „modern” natív játékok között több csoportot vélhetünk felfedezni. Vannak a platformtól független kereskedelmi megvalósítások, vannak aztán

a – sok esetben – idegen kezek által átültetett kereskedelmi projektek, de akadnak olyan szabad játékok is, amelyek (általában *GPL*) forráskódja gyakorlatilag az összes, tényleges felhasználói táborral rendelkező operációs rendszerre felkészíthető és azon lefordítható.

A legjellemzőbb független kereskedelmi játékokat mind a mai napig az előzőekben már említett *id Software* fejlesztői adják ki kezeik közül. Rugalmas kódjaiknak köszönhetően valójában a konkurenciát is belekényszerítve az említett függetlenség követésébe. A „*Linux*-képes” kereskedelmi játékok nagyágyúit nem hiszem, hogy be kellene mutatnom, elég lesz csupán a *Doom/Quake* sorozatra, az *Unreal Tournament* ágra, vagy a *Neverwinter Nights* családra gondolni: mára már nem kell megérőltetnie magát annak, aki ebbe a kategóriába tartozó népszerű játékot szeretne említeni.

Ezek a programok jellemzően *OpenGL API*-n keresztül programozzák a grafikai hardvert, ahol gyakran az *SDL* könyvtárrendszert is segítségül hívják. Hangszolgáltatásuk *ALSA/OSS* meghajtóra épülve sokszor *OpenAL* illetve *FMod* könyvtárrakra támaszkodik, mely leképezési módszerek többek között *Linux*, és *Win32* operációs rendszereken egyaránt elérhetőek. Így a fejlesztőknek effektíve könnyű dolga akad az eltérő rendszerekre fordított binárisok felépítését illetően, miközben arról sem szabad megfeledkeznünk, hogy igen hatékonyan kihasználható, jól

dokumentált és „masszív” alapokról beszélünk, melyek már számos helyen bizonyították: van élet a *DirectX*-en túl.

A második csoport

Ez a csoport kitüntetett figyelmet érdemel, mivel az a „porter”, aki már egy elkészült, *Win32/DirectX*-re támaszkodó kódot szeretne átültetni *Linuxra*, jellemzően olyan komoly feladat elé néz, melynek kivitelezése elképesztő (és sokszor „önzetlen”, anyagiaktól mentes) munkával jár. Ezen a területen feltétel nélküli tisztelettel tekinthetünk a *Scott Draeker* vezette, néhai *Loki Entertainment* munkásságára, hiszen ők voltak azok az úttörők, akik a *DirectX* környezetre fejlesztett játékok magját tömegesen írták át a szabad rendszerre, jellemzően *OpenGL* és *SDL* programozási felületekre / rétegekre támaszkodva.

Közel húsz, igen komoly potenciállal rendelkező átültetés fűződik nevükhöz. Nem sorolom őket, de aki kíváncsi a listára, az a még elérhető (és sajnos négy éve inaktív) <http://www.lokigames.com> oldalon minden témába vágó információt megtalál. A csapat hiánya mind a mai napig érezhető, bár tevékenységük természetesen maradandó és számmunkra felbecsülhetetlen értékű: az olyan nagyágyúk, mint a *HOMM3* vagy éppen a *Descent3* a *Loki* nélkül nem léteznének natív, linuxos verzióban. (Szintén figyelemre méltó az általuk megálmodott, jellemzően *.run formájú telepítő alkalmazás, mely mindmáig használatban van, egyéb projektek által.) A *Loki* szerepét ma leginkább a nem kevésbé szakértő *Icculus* csapata tölti be, több – kevesebb sikerrel, eltérő (nem kereskedelmi) szemlélettel.

A <http://icculus.org> oldalon hosszú listán keresztül követhető nyomon a párhuzamosan foltozott kódok állapota, nem ritkán bináris formában és *Loki*-alapú telepítőbe drótozott elérhetőséggel. Természetesen azokról a „kisebb” porterekről se feledkezünk meg, akik néhány fős csapatokkal vállalkoznak a nehéz feladatra. Jó példa erre a *Relic Homeworld*-jét átdolgozó csoport, hiszen ennek linuxos változata is meggyőzőre sikerült az idegen kezek által. Ennél

a „műfajnál” a legfőbb buktatót valójában maga az eredeti fejlesztő jelentheti: amennyiben bármilyen okból kizárólag a piacvezető operációs rendszert támogatja, teljesen érthetően közel sem biztos, hogy programjának fő kódját belátható időn belül külsős csapatra bízna – nem beszélve arról a nehézségről, ha ez a kód veletlenül *DirectX* specifikus. Szerencsére bőven akad életképes ellenpélda, a kereskedelmi szemlélettel dolgozó „porterek” (*Loki*, *Hyperion*) anyagain felül példaként említhető az előbb említett *Homeworld*, de akár a *FreeSpace2*-t is megnevezhetem (mely az *Icculus* egyik projektjeként változott át). Utóbbi esetekben köszönettel tartozunk a fejlesztőknek és kiadóknak is, a forráskód önzetlen megnyitásának ügyén. Rendkívül érdekes azt a folyamatot megfigyelni, ahogyan az átirított játékok lassan, de biztosan szaporodnak: bár a *Loki* bő négy évvel ezelőtti bezárása nem volt túlzottan jó előjel, a jelen történései úgy gondolom biztatóak. A reményteljes folyamat kimenetele nagyon sok dologtól függ, többek között természetesen attól is, hogy az elkövetkező néhány évben a szabad rendszerek az asztali szegmensben milyen részesedésre lesznek képesek szert tenni. Az sem mellékes, hogy a *Win32/DirectX* rendszerkörnyezet, és ennek marketing munkálatai milyen eséllyel lesznek képesek a programozók zömét továbbra is kizárólagosságra készíteni (sajnos ez a lehetőség még mindig valós veszélyforrásként lebeg előttem).

A teljes szabadság

A harmadik kategória cikkem írásakor (múltjához mérten) már egy másik „dimenziót” kóstolhat: az olyan szabadon elérhető projektek, mint a *Glest*, a *Racer*, vagy a *FlightGear* az idő múlásával mind komolyabb alternatívát fognak jelenteni a játékok között, mely megoldások jelenléte (előbb vagy utóbb) akár a két előző csoport ellenében is érezhető lesz. Ebben a csoportban a kisebb, egyszerűbb próbálkozásokon keresztül jó néhány olyan elképzelés is fellelhető a világhálón, melyeket az előbb példaként említettem, és amelyek állapotukat tekintve már most sokkal többet ígérnek, mint egy átlagos

platformfüggetlen „másolat”. Mivel eme régió megvalósításainak programkódja gyakran mindenki által elérhető, így nem ritkán a publikálást követően akár *OS X*-en futó bináris verziók is felbukkannak néhány nap elteltével, jellemzően szakértő „kezek” által alakítva.

Mit hozhat a jövő?

Nem győzöm hangsúlyozni, a rejtély kulcsa a platformfüggetlenség. A cikk írása előtt néhány nappal töltöttem le a néhai *Quake2*-ből készített *Jake2*-t, melyben fő változásként az egész játék *Java* értelmezőn fut – így gyakorlatilag minden rendszeren működik, és mindenütt ugyanazt nyújtja. Mindezt a *Quake2* nyitott forráskódjából készítette egy német csapat, olyan önzetlen munkát végezve, ami által már nehéz olyan asztali operációs rendszert mondani, ahol e népszerű játék módosított kódja ne futna (holott az *id Software* gyermeke már eleve multiplatformos kód, bár nem ennyire széles használhatósággal). A jövő remélhetőleg a fejlesztők ez irányú felismeréséről fog szólni, mivel független kódjaikkal saját piacukat bővíthetik tovább, ami minden bizonnyal mérhető profittal (és szakmai elismeréssel) kecsegtet. Márpedig a *Linux* (vagy éppenséggel a *BeOS*, *OS X*) az otthoni, asztali gépek piacán mind gyakrabban köszön vissza. Mindezen felül rendkívül érdekes megfigyelni a játékprogramok és az asztali *PC* szegmens közötti kölcsönhatást is: a *Linux* (és az egyéb alternatívák) mind erősebb terjedése nagyobb platformfüggetlenséget szül, miközben a független játékok megkérdőjelezhetetlenül segítik a rendszer asztali elterjedését. Személy szerint bizakodó vagyok: habár az alagút végét esélyem sincs meglátni, de részemről bizalmat szavazok annak a közösségnek, mely szabadon elérhető forráskódú játékokat készít nekünk, vagy éppen a mind hatékonyabb függetlenség modelljén dolgozik.

Kovács Zsolt (kovi@linuxforum.hu)

Quake fanatikus. Négy éve a debreceni linuxosok egyike. Töretlenül hisz a Slackware terjesztésben.