

NVClock – nVidia tuning Linuxon

A Linux-felhasználók számítógépeiben általában nVidia grafikus hardver lapul. Nem véletlenül, hiszen a favorit gyártó köztudottan a konkurencia előtt jár, a számunkra fontos támogatás terén. A 3D eszközeik finomhangolásról és a túlhajtásról azonban méltánytalanul kevés szó esik...

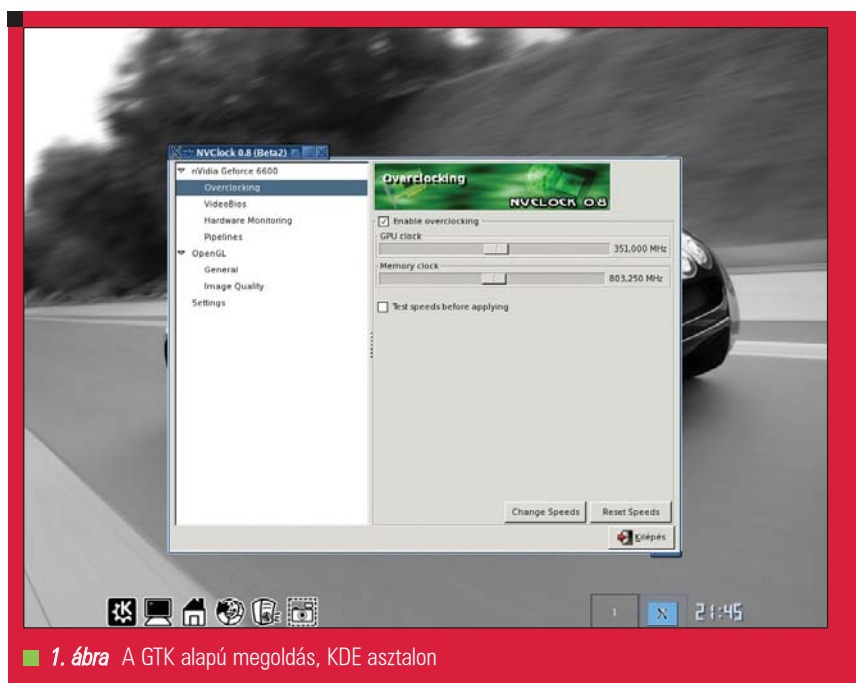
nVidia, és megint csak nVidia

Ha valaki komolyabb 3D grafikus munkát végez Linuxon, vagy éppen OpenGL programozási felületre támaszkodó játékokkal játszik, akkor bizonyára egyetért a bevezetővel. Akár alaplap csipkészletbe égetett grafikus vezérlőről, akár PCI-Express / AGP sínbe helyezett kártyáról legyen szó, az nVidia hardvereit nagyjából két perc alatt rá lehet bírni a gyári meghajtók használatára – ezáltal a népszerű 3D leírnyelv ismeretére is. Nem mintha a konkurencia képtelen lenne kielégítő támogatást nyújtani megoldásaihoz, de a cikk írásakor összességében (hatékonyság, kényelem és kompatibilitás terén) én is mérhetőnek érzem az említett gyártó linuxos népszerűségi fölényét. Szerencsére a GeForce osztályú grafikus hardvereket nem kell különösebben bemutatnom az Olvasónak.

Hiszen az ide vonatkozó szilícium-technikai újdonságokat az Internet segítségével bárki könnyedén nyomon követheti, a telepítés és a használatba vétel témája pedig már többször szerepelt e magazin hasábjain is. Az újabb meghajtók installálási módja pontosan egyezik az előző generációban megismertekkel, a kódok változási listája pedig a gyártó honlapján elég jól van dokumentálva. Ezért ezt a részt hanyagolni fogom. Ellenben a cikk teljes megértéséhez minimális szakmai ismeretet fogok feltételezni.

A feladat felelőssége

Szinte biztosan akad olyan érdeklődő, aki már összefutott a következő jelen-



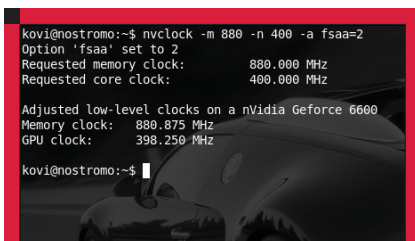
1. ábra A GTK alapú megoldás, KDE asztalon

séggel: egy bizonyos szoftver (leginkább egy nagy igényű játékprogram) adott számítógépen nem fut kielégítő sebességgel. Ennek a kellemetlen helyzetnek több oka is lehet, kezdve a CPU gyengeségétől, a memória szerény méretén át egészen valamely szűkös „adat-keresztmetszetig”. Természetesen az okok között szerepel a grafikus vezérlő elégtelensége is, ami a cikk szempontjából kiemelt fontosságú. Hogy miért? Utóbbi esetben a tehetősebbek azonnal megrohanják a boltok polcait (egy modernebb darabért), mások benyugszanak a lehangoló helyzetbe, de vannak, akik különböző trükköket vetnek be orvoslásképpen...

Ezek a „trükkök” jellemzően arra épülnek, hogy a minőségi munkákról elhíresült, neves kártyagyártók termékeiben sokszor kihasználatlan (de kihasználható) teljesítménytöbblet lapul. A feladat adott, csalogassuk hát elő a kiaknázatlan lehetőségeket! Mielőtt azonban belevágnék a téma ismertetésébe, két dolgot le kell szögez-nem. Az előbb írt teljesítménytöbblettől nem szabad csodát várni. Ha egy program a hardver alapvető képességeinek hiányában nem élvezhető, akkor nincs értelme sem finomhangolással, sem pedig tuninggal próbálkozni. A másik dolog amire ki kell térjek, annak felelősségteljes hozzáállásra és ésszerű megoldások keresésére kell ösztönöznie az

1. táblázat *Az NVClock használata*

<code>nvclock -m</code>	A kártyán lévő memória órajelének beállítása (MHz)
<code>nvclock -n</code>	A grafikus mag sebességének beállítása (MHz)
<code>nvclock -r</code>	Az eredeti értékek visszaállítása (BIOS alapján)
<code>nvclock -s</code>	A jelenleg érvényes órajelek megjelenítése
<code>nvclock -d</code>	Nyomkövetési (debug) információk visszajelzésének kapcsolása
<code>nvclock -c</code>	Az állítandó kártya száma (többkártyás rendszereknél)
<code>nvclock -f</code>	Kényszerített (force) mód, vagyis a hangolás kikényszerítése
<code>nvclock -F</code>	A hűtőventilátor szabályozása (10-100%)
<code>nvclock -P</code>	Az extra pixel-csatornák bekapcsolása (ha van)
<code>nvclock -Q</code>	A PCI buszazonosító felülbírállása
<code>nvclock -S</code>	A smartdimmer fényerő szabályozása (15-100%)
<code>nvclock -T</code>	A grafikus mag hőmérsékletének kijelzése
<code>nvclock -i</code>	A hardver részletes információinak kijelzése
<code>nvclock -a</code>	Az nVidia meghajtó opcióinak beállítása (pl. FSAA)
<code>nvclock -q</code>	Adott nVidia opció lehetséges értékeinek kijelzése
<code>nvclock -l</code>	Használható nVidia / OpenGL opciók kijelzése
<code>nvclock -x</code>	Másik X megjelenítőre váltás



2. ábra A konzolos verzió használata

Olvasót. Arról van szó, hogy egy olyan videokártyát, melynek a központi egységét nem ritkán 20 millió tranzisztor építi fel, oktalan és szakszerűtlen „beavatkozással” pillanatok alatt a másvilágra lehet küldeni. Márpedig a leégett, elfüstölt, vagy éppen csak elszíneződött alkatrészek a garanciális ügyintézés (szinte) minden üzletben kizárják. Röviden összefoglalva az intelmeket: a továbbiakban leírt dolgokkal csak ésszerű keretek között, saját lehetőségeihez mérten és várható eredmények reményében próbálkozzon bárki.

Az NVClock felélesztése

A Win32 környezet felhasználói bőségesen el vannak látva mindazokkal az eszközökkel, amivel egy nVidia grafikus hardver teljesítménye emelhető. A megoldások számát tekintve Linux alatt már kevésbé jó az ellátottság,

szerencsére azonban akad olyan Szabad projekt, mely letisztult, lényegre törő és hatékony – ezzel el is jutottunk az NVClock-ig... A szoftver honlapja a <http://www.linuxhardware.org/nvclock/> URL mögött található, innen érhető el a GPL licenc szerint terjesztett forráskódja.

Töltsük le az `nvclock_verzió.tar.gz` tarballt, majd a kicsomagolt archívban adjuk ki a szokásos

```
./configure, make, make install
```

parancsokat (az utolsót természetesen root jogkörrel kell megtenni). Ennek hatására felépülnek és helyükre (`/usr/bin`) kerülnek az ELF binárisok, melyek közül az egyik a konzolos futásért, a másik a GTK alapú GUI-val ellátott használatért felel. A szöveges felület hívei ezután a felhasználóként kiadott

```
nvclock -
```

a hangolás grafikus lehetőségét szorgalmazók pedig az

```
nvclock_gtk
```

paranccsal juthatnak közelebb a kész megoldáshoz (feltéve, hogy a konfiguráló szkript talált használható GTK

könyvtárat). Fontos, hogy az NVClock teljes körű funkcionalitása egy használható (induláskor betöltött) gyári nVidia modult feltételez a rendszerünkben!

És a gyakorlat...

Akár a grafikus interfésszel ellátott verziót, akár a terminálon használható megoldást választja az érdeklődő, nagyon egyszerű használatot kell észben tartania. A program három célt szolgál: először is a grafikus hardverről ad részletes információkat, másodsor lehetőséget biztosít ennek finomhangolására, végül pedig (de nem utolsó sorban) a túlhajtásra fókuszál. A fő opciókat a konzolos megoldásnál úgy lehet életre hívni, hogy az `nvclock` parancsot felparaméterezzük a mellékelt táblázat szerint. A GTK alapú ablakos megoldás nem szorul különösebb magyarázatra, az ide kapcsolódó ábrából reményeim szerint látszik a projekt lényegre törő felépítése.

A program nem képes végleges változásokat kieszközölni, ezek a beavatkozások továbbra is csak a ROM BIOS módosításával érhetőek el. Ennek ellenére bizonyos értelemben alkothatunk maradandót, hiszen az NVClock a Linux induláskor automatikusan betölthető (a megfelelő init szintek módosításával, vagy esetemben akár a KDE indító szolgáltatásával). A beállított paraméterek mindig a megfelelő `/home/$/.nvclock/config` állományban vannak lerögzítve, tehát a kívánt értékek az aktuális felhasználói profiltól is függhetnek. E fájlok kézi szerkesztésétől óva intek mindenkit: bizonyos sorokban, a nehezen értelmezhető értékek megváltoztatásával könnyen előidézhető az adott hardver totális leégése. Ezek után nézzük az életszerű példát! Nekem egy Asus márkájú, Top Silencer videokártyám van: ez annyit tesz, hogy a nyák ellentétes oldalára került a grafikus processzor (esetemben egy középszerű GeForce 6600). A mag így kevésbé melegszik, hiszen (álló házba építve) a termelt hő felfelé szabadon távozhat. Az sem mellékes, hogy ez a típusú 6600-as amúgy sem melegezős, a ventilátor helyett egy hatalmas radiátor felel a hangtalan hűtésért. Az Asus konzervatív és profi politikájára jellemzően a válogatott grafikus mag gyári órajelen fut, mint ahogyan a felhasznált memória is hivatalos ajánlás alapján van időzítve – miközben

nagyobb sebességen is hibátlanul működik. Kiadom tehát egy konzolon az

```
nvclock -m 880 -n 400 -a fsaa=2
```

parancsot, ami által a grafikus mag sebességét (a memóriához hasonlóan) a gyári értékek fölé emelem 15%-al. Mindemellett a példában lebutítom a teljes képernyős élsimítás képességét is (a mintavételezési érték rögzítésével), így a 3D szoftverből történő automatikus vezérlésnek is keresztbe teszek. Az eredmény: közel 10 Celsius fokkal melegebb GPU, közel 15%-al gyorsabban futó Unreal Tournament 2004 (ebben a 15%-ban természetesen érződik az FSAA kapcsolása is, mely 1024x768 felbontásban nagyjából 5 százalékot segít nekem). Persze játék végén nvclock -r paranccsal visszaállítok mindent az eredeti állapotba. Ha a GTK alapú megoldást választom, ez a teljes folyamat öt kattintás árán kivitelezhető...

Figyelem!

Nem győzöm hangsúlyozni: minden változtatás kiemelt figyelemmel hajtható végre! Egy csendesre állított

(alacsony fordulátú) ventilátor, egy túl gyorsra állított mag pillanatok alatt komoly anyagi kárt képes okozni. Hiszen a GPU tranzisztoraiból, vagy a memória celláiból csak egynek kell tönkremennie, és máris kész a baj.

Szerényebb minőségű hardvereken inkább ne eszközöljünk semmilyen változtatást, hiszen egy (válogatás útján) gyengébbnek ítélt grafikus mag elemei általában nem viselik zokszó nélkül a gyorsabb nyitástartást. Abban az esetben lehet érdekes az ismertetett program, ha éppen annyi teljesítmény többlet hiányzik egy játék élvezhető futásához, mint amennyi kisajtolható egy jobb fajta nVidia alapú kártyából – ésszerű keretek között. Mindemellett az esetlegesen bekövetkezett balesetekért a felhasználó az egyetlen felelős: sem a gyártók, sem pedig az NVClock fejlesztői nem vállalják fel ezt a „tisztiséget” (teljesen érthető módon én sem tarthatom a hátam az említett példa személyre szabott sikeréért). Így a program README állományát minden leendő használó olvassa el.

Vélemények

A túlhűzés mértékéről és létjogosultságáról megoszlanak a vélemények. Van, aki igazi megszállottként próbálja sebesebb tempóra kényszeríteni hardverét, de akad olyan is, akit kiráz a hideg ettől a dologtól. Nem célokom állást foglalni a kérdésben, de a magam óvatos megközelítésével sokkal inkább a középutat keresem: ritkán hajtom túl a (lassan kiöregedő) grafikus kártyámat. Egy dologgal azonban már én sem vagyok kibékülve: az előbb említett első csoportból néhányan sportot űznek ebből a „játékból”, és akkor is tuningra adják a fejüket, ha igazából nincs is szükségük rá. Ha valaki legalább ezt az egy kapitális hibát igyekszik elkerülni, akkor már meg is tette az első lépést az NVClock értelmes használata felé. Sikeres, és főként „balesetmentes” próbákat mindenkinek!

Kovács Zsolt (kovi@linuxforum.hu)

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



© Kiskapu Kft. Minden jog fenntartva