



Digitális mozgóképsorok saját kezűleg (2. rész)

MEncoder – kikövezett út a hatékony tárolás felé, ismét megújult formában.

Legutóbbi írásom végén azt ígértém, hogy ebben a részben a gyakorlaté lesz a főszerep, ami természetesen igaz is, azonban mielőtt belevetnénk magunkat a sűrűjébe, meg kell vizsgálnunk az előző cikkben leírtakat. Ennek az az oka, hogy a cikk megírása óta az MPlayer (vele együtt a MEncoder is) látványos változásokon ment keresztül, ami rendkívül kedvező fordulat, ugyanis rengeteg új lehetőség és javítás került bele, ám ezzel együtt némiképpen a kezelés módja is megváltozott.

Nem kell megijednünk, hiszen az alapelgondolás maradt a régi, csupán az egyes kapcsolók és értékek megadásának módja változott néhol, feltehetően a könnyebb vagy inkább strukturáltabb használat és felépítés érdekében.

A fejlesztés tehát gőzerővel folytatódik, újabb és újabb kiadások jelennek meg (a cikk írásának pillanatában a 0.9pre2 változatnál tartunk, ám elképzelhető, hogy holnapra már ez is „elavult” lesz). Ez a „gőzerő” a csomag statisztikáin meg is látszik. Az egyik legnagyobb linuxos weboldalon, a Freshmeaten (<http://www.freshmeat.net>) jó néhány kimutatás található, köztük a legnépszerűbb projektek húszas toplistája is, amelyen az első helyet a rendszermag foglalja el 100 százalékponttal, ezt követi az MPlayer 95,5 százalékponttal (összehasonlításképpen: a 3. helyezett cdreord csak 47 százalékpontot szerzett). Azt hiszem, a számok önmagukért beszélnek, a program igen kedvelt, így hát mi is tanulmányozzuk tovább!

Néhány szó a legújabb fejlesztésekről

A cikk terjedelméből kifolyólag nem lehet minden egyes újítást, javítást, kényelmi megoldást felsorolni, néhányat azonban ki szeretnék emelni közülük. Az egyik az operációs rendszerektől független egyedülálló feliratozás. Nagyszerűségét abban látom, hogy míg ablakos nézetben az árnyékolt feliratot vagy órát a filmre helyezi, addig teljes képernyős módban a képernyőn lévő holttérre próbálja meg helyezni. A holttér a monitor (4:3) és a film (16:9; 2,25:1) képarányának eltéréseiből adódik.

Léteznek olyan feliratozók is, amelyek a feliratot ablakban képesek megjeleníteni, ám növelik az ablak méretét (4:3 képarány), hatalmas, kétszerakkora helyet foglalva el a képernyőből. Ha valakinek mégis ez tetszik, az Expand nevű szűrővel ablakban is készíthet magának holtteret. A másik vonzó javítás, hogy a lejátszómotor és a kodekek fejlesztésének köszönhetően a lejátszási sebesség is javult. A lassúbb masinával rendelkező felhasználók ezt leginkább számításigényes DVD-filmek lejátszása során érezhetik.

A legutóbbi cikk óta kialakítottak egy videoszűrő réteget (ez az egyik oka annak, hogy a paraméterezés megváltozott), amely ugyanazt a munkát végzi feltételezhetően ugyanazon a módon, ám sokat egyszerűsít a kezelésem és annak megértésén, egyben felkészülés is a további képkezelő szűrők fogadására. Az új változat több ilyenrel is bővült, ezek közül az egyik az előző írásom végén hiányolt képszegélylevágás (cropping) (a folyamat fordítottja is létezik, amit az Expand szűrő valósít meg). A régi videoszűrő eljárások (flip, postprocess) természetesen a programban maradtak, csak ezúttal a -vop kapcsoló után felsorolva lehet megadni őket. Az új változatban már a libavcodec kodekcsalád DivX-tagjával is lehetőség nyílik a kétmenetes kódolás alkalmazására. Ezekkel a megoldásokkal gyakorlatilag egy profi DVD-„átalakító”, DivX-es AVI-készítő eszköz áll rendelkezésünkre.

Vissza a MEncoderhez

Kezdjük rögtön a változások ismertetésével! Javaslatom, hogy mindenekelőtt a 0.9 pre2 változatot telepítsük (34. CD/Magazin/MPlayer könyvtár), hogy az itt leírtak ne térjenek el attól, amit olvasóink hiba nélkül alkalmazni tudnak. A telepítés folyamata teljesen megegyezik az MPlayerről szóló cikkünkben leírtakkal (Linuxvilág 2002. március, 28. oldal). Azt csiripelik a kismadarak, hogy hamarosan megjelenik a pre3 változat, amelyet már előrefordított csomagok formájában is el lehet érni, így a hosszas telepítési, fordítási folya-

matot megtakaríthatjuk magunknak. A legfontosabb változás az egészséges videoszűrő réteg bevezetése (a továbbiak az MPlayerre is igazak, a lejátszás szintén ily módon zajlik). Ha a MEncoder kimeneti képtulajdonságait szeretnénk megadni, a -vop kapcsoló mögött kell őket felsorolni, egymástól vesszővel elválasztva. A sorrend nem közömbös! A program a kapcsolólistában szereplő szűrőket fordított sorrendben, a folyamatban hátulról visszafelé fűzi láncba. Rossz sorrend esetén előfordulhat, hogy nem azt kapjuk, amit szeretnénk, néha a MEncoder nem is tud mit kezdeni a hibás értékekkel, és futása hibával megszakad. Ilyen például, amikor a kép egy adott részét szeretnénk kivágni és más nagyságúra átméretezni. Ebben az esetben, ha a kivágást előbb adjuk meg – ami azt jelenti, hogy a folyamat először az átméretezéssel fog kezdeni –, meglehet, hogy az átméretezett képből már nem onnan és nem ugyanakkora darabot vág ki a képszegélylevágás művelet, mint amit mi eredetileg szeretnénk volna. A képszegélylevágás művelet a másik örömmel fogadott újítás. Sokszor a bemeneti eszköztől (ez leggyakrabban a DVD-lemez) nyert képeknek van egy vékony fekete kerete, sőt, az is előfordul, hogy a 16:9 arányú kép 4:3 dobozban jelenik meg, és a maradék terület fekete. A holttér tárolása nemcsak helyvesztéséget okoz, de a lejátszásnál többlet-processzoridőt igényel, azonkívül nem is esztétikus. A program eme lehetőségét a -vop crop=`<0rt0kek>` kapcsolóval használhatjuk. A képszegélylevágás utáni értékek helyére először a kivágás szélessége, magassága, ezután a kivágás vízszintes, majd függőleges kezdőhelye kerül, egymástól kettőspontokkal elválasztva. A kimeneti kép átméretezését a korábbi -x, -y kapcsolók helyett szintén a -vop kapcsoló után felsorolandó `scale=<x>:<y>` értékekkel adhatjuk meg, ahol az x és y értéke a kép szélességének és magasságának felel meg. Ezek a leggyakrabban használt szűrők, de bőven akadnak mellettük mások is. Ezekről az MPlayer leírásának idevágó szakaszában olvashatunk részletesebben.

Kódoljunk DivX-et!

Ez a feladat már korántsem olyan látványos, mint a lejátszás. Kép helyett csak állapotjelző értékeket látunk, ráadásul rendkívül lassú. Az eljárás egyes gépeken a lejátszási sebességnél talán egy picivel gyorsabban is működik, ám a legtöbb esetben kétmenetes kódolást alkalmazunk, ami a tömörítési időt a kétszeresére növeli. Erre a jobb minőség elérése érdekében van szükség. Mint már említettem, a videotömörítés alapja az, hogy az egymás után következő képek csak kis mértékben különböznek egymástól. Ezt kihasználva egyenletes mozgásnál szép eredményeket érhetünk el. Ezzel csak az a baj, hogy a mozgás általában nem egyenletes, ezért a gyors mozgások esetén a kép eltorzulhat, és romlik a minősége. Úgy védekezhetünk ellene, ha változó bitrátájú tömörítést alkalmazunk, amely a gyors mozgásoknál kevesebb veszteséggel dolgozik, így azonos képminőséget mutat fel, mint a lassúbb mozgások. Ehhez azonban tudnunk kell, hol vannak a gyors mozgások. A másik gond a tömörített méret kiszámítása: ha ugyanis változó tömörítési mértéket alkalmazunk, a kimeneti fájl méretét még csak becsülni sem tudjuk. A kétmenetes tömörítési eljárás mindkét nehézséget megoldja, ez első lépésben állandó bitrátával tömöríti a filmet (ezt természetesen nem használjuk), közben elemzi a mozgásokat és naplózza őket. A következő lépésben újra elejéről kezd a tömörítést, és az előző lépés során készített naplófájl alapján gyakorlatilag az egyes pillanatokban alkalmazott tömörítési arányt úgy „csoportosítja át”, hogy a lassúbb mozgásoktól „elveszi”, és a gyorsabbaknak „adja”, így végeredményben ugyanakkora méretű, ám a gyors mozgásoknál is jó képminőséget biztosító eljárásához jutunk. A MEncoder hárommenetes tömörítésre is képes, amely egy olyan kétmenetes eljárás, amelyben a hangsáv külön lépésben kerül az AVI-ba. Ez az első lépés. Előnye, hogy mivel a hangsáv mérete és hossza ismert (feltételezzük, hogy a hang éppolyan hosszú, mint a film), így a program ennek a lépésnek a végén a kívánt adathordozó méretéhez (650, 700, 800 megabájt) viszonyított képsáv tömörítésének mértékét ajánlja fel számunkra (több CD esetén a kapott értéket szoroznunk kell). Ezután következhet a másik két lépés.

DVD-ből AVI

Előfordulhat, hogy szerényebb méretben másolatot kell készítenünk egy filmről. Ha MEncodert használunk,

erre nagyon egyszerűen lehetőségünk nyílik. A lejátszáshoz hasonlóan módon, 1–2 további kapcsolóval és némi processzoridővel (10–12 óra, esetleg több) segíthetünk magunkon. Mik a teendők? Először is a filmet vissza kell nyernünk az adathordozóról, amit a program elvégez helyettünk. A következő lépés a keret levágása. Ehhez tudnunk kell a film fizikai méretét (PAL-szabvány esetén 720×576), amelyet lejátszáskor az MPlayer (a MEncoder szintén) a konzolra ír. Ezek után az MPlayer segítségével meg kell határoznunk, hogy mekkora részt és honnan kell kivágni a képből. Ezt úgy tehetjük meg, hogy mentünk (capture) egy képet, majd szerkesztőprogram segítségével keretet húzunk arra a részre, amit látni szeretnénk, ezt követően megnézzük az értékeit. A másik lehetőség: az MPlayer segítségével a látni kívánt részt nézegetési próbákkal határoljuk be. Ezek után át kell méreteznünk a képet, hiszen a legtöbb esetben a fizikai felbontás nem a helyes képarányt tükrözi (lásd Linuxvilág 2002. április, 48. oldal), és sokszor a DVD-n alkalmazottnál kisebb felbontással is megelégszünk. Ezt a `-vop scale=<szólesség>:<magasság>` kapcsolóval adhatjuk meg. Az értékeket az eredeti képarány alapján számológép segítségével célszerű kiszámolni. Ezek után meg kell adnunk a kép és a hang formátumát, majd kétszer (háromszor) le kell futtatni a tömörítést.

Vegyünk egy példát, amelyben hárommenetes kódolást alkalmazva megtartjuk az eredeti ac3 hangot, és tömörítésre a divx4 kodeket használjuk 1400 kbit/s tömörítési aránnyal. Az eredeti anyag 720×576-os, melynek a kép szélein 8 pixel, a tetején és az alján pedig 128 pixel széles fekete keret található (ebben az esetben a látszólagos és a fizikai képarány megegyezett). A kimeneti kép 640×272-es felbontású lesz. A fenti eredmény eléréséhez az alábbi parancsokat kell kiadni.

```
1. mencoder -dvd 2 -ovc frameno
   ↪ -oac copy -o frameno.avi
   //a kimeneti fájl neve
   k telezi!
```

Ennek hatására létrejön egy kép nélküli AVI, amely a film hangját tartalmazza, az alkalmazandó tömörítési arányt (1400 kbit/s) pedig eredményül a futás végén kapjuk meg.

```
2. mencoder -dvd 2 -pass 1 -vop
   ↪ scale=640:272,crop=712:320
```

```
↪ :8:128 -oac copy -ovc divx4
↪ -divx4opts br=1400 -o
↪ mymovie.avi
//A program tudja, hogy a
hang a frameno.avi fájlban
található .
```

Egy AVI-t (ez most lényegtelen) és egy *divx2pass.log* nevű fájlt kapunk, amely a mozgáselemzés eredményét tartalmazza. Ez az, ami fontos.

```
3. mencoder -dvd 2 -pass 2 -vop
   ↪ scale=640:272,crop=712:320
   ↪ :8:128 -oac copy -ovc divx4
   ↪ -divx4opts br=1400 -o
   ↪ mymovie.avi
```

Ez az utolsó lépés. A program mind a *frameno.avi*, mind a *divx2pass.log* fájlokat érzékeli, és ezek alapján elkészíti a végleges filmet.

Az itt felsorolt értékeket előre kiszámoltam, sajnos ezek minden lemez esetében eltérőek, mindig újra kell őket számolni, majd célszerű legalább a videoszűrőkre vonatkozókat MPlayerrel ellenőrizni. Még oldalakon át mesélhetnék a program további képességeiről, lehetőségeiről és az alkalmazható huncutságokról, ám azt hiszem, egy fél magazin is kevés lenne, ha mindent le szeretnék írni, épp ezért minden érdeklődőnek figyelmébe ajánlom a magyar nyelvű leírást, melynek tanulmányozásával és rengeteg „gyakorlással” tapasztalatokat szerezhettek a videotömörítés nemcsak linuxos, de általános fortélyairól is, miközben a programban rejlő számtalan apró lehetőséget is elsajátíthatják. Az MPlayer törtetlen fejlődését és egyre nagyobb népszerűségét elnézve, azt hiszem, lesz még miről írnom a továbbiakban.



Komáromi Zoltán

(komi_@freemail.hu)
21 éves, a BME hallgatója, mellette PHP-programozóként dolgozik.

Kedvenc területe a multimédia. Kedveli a nagy társaságot, az érdekes embereket, a jó filmeket és mindent, ami mozgalmos. Szabadidejében röplabdázik.

Kapcsolódó címek

- <http://www.MPlayerHQ.hu/homepage/>
- <http://www.mplayerhq.hu/DOCS/Hungarian/documentation.html>