

RTcmix a Linuxon (1. rész)

A valós idejű hangképzéssel foglalkozó sorozat első részében az RTcmix történetével és alapjaival ismerkedhetünk meg.

Valós idejű hangképzést és -feldolgozást tesz lehetővé az RTcmix nevű programcsomag. Tulajdonképpen ez nem más, mint a Cmix nyelv valós idejű változata. A program futását egy MINC (MINC Is Not C) nevű parancsnnyelvvvel vezérelhetjük. Ez már akkor is nyílt forrású volt, amikor e kifejezés csak kevesek számára jelentett valamit.

Története

Az elmúlt húsz év alatt több vállalkozó kedvű programozó is hozzátett valamicskét az RTcmixhez. Írásomban megkísérlem minden érintett nevét felsorolni, ám ha valakit mégis kihagytam volna, a csomagban található AUTHORS fájl tartalmazza mindet. A Cmix a MIX-ből fejlődött ki, ezt húszsávú keverőprogramból írta *Paul Lansky* a princetoni egyetemen. A VMS-alapú IBM nagygépekre tervezett eredeti programot 1978-ban Fortran nyelven írták. Paul a hangképzési képességekkel bővített programot 1983–84-ben átírta BSD 2.9-es Unixra. 1985-ben a fejlesztés egy Ultrixot futtató DEC MicroVAX-ra költözött át. A Cmixet a továbbiakban NeXT, illetve más Unix rendszereken lehetett elérni, és a legtöbb felhasználója az egyetemi körökből került ki. Már akkor is főleg zeneszerkesztésre és zenei kutatásokra használták. 1995-ben *Doug Scott* és *Paul Lansky* az első valós idejű változatot átírta SGI-re. Én 1993-ban, a Columbia egyetem hallgatójaként találkoztam először a program linuxos változatával, amely még Soxot használt a fájlkezelés megkönnyítésére. Azok voltak csak a szép idők... 1995 végén *Brad Garton*, az egyetem számítógépes zenetanszékének (CMC) vezetője és jómagam is készítettünk egy valós idejű változatot, ütemezővel együtt. Az eredeti RTcmix IRIX alatt futott. Átültetése Linuxra annyit jelentett, hogy az IRIX hangkezelő rutinjait OSS (Open Sound System) eljárásokra cseréltük le. Az első IRIX-változatot az 1997-es International Computer Music Conference keretein belül mutatták be. Röviddel ezután a CMC másik vezetője, *Luke DuBois* a TCP csatolóadatok írását megkönnyítő eljárásokat épített bele. *Doug Scott* a hangszerek dinamikus, futásidőben történő betöltését már 1997-ben megvalósította. *John Gibson* (*Bill Schottstaedt* sndlib-jének felhasználásával) különböző hangfájl-formátumok olvasásának és írásának lehetőségével bővítette a programot. *Johnnal* nemrégiben

kidolgoztunk két újítást: a hangadatok belső „buszon” történő továbbítását, illetve a többcsatornás hang támogatását. Munkánk eredményét a SEAMUS (ez egy számítógépes zenével foglalkozó társaság) idei gyűlésén mutattuk be. További terveink között szerepel a hang-

1. lista WAVETABLE_REVERBIT_1.sco

```
/* WAVETABLE_REVERBIT_1.sco */
/* Megmutatja, hogyan illeszthetjük be a visszhanghatást bármilyen
   kottába. */
rtsetparams(44100, 2)
load("WAVETABLE")
load("REVERBIT")
/* Ha fájlba akarunk írni, töröljük a következő két sorból a
   megjegyzésjeleket. */
/* set_option("CLOBBER_ON") */
/* rtoutput("wavetable_reverbit1.aiff") */
/* A WAVETABLE kimenete az aux 0-1 buszra érkezik, ezt olvassa be a
   REVERBIT, mely a feldolgozott adatokat a hangkártyára küldi. */
bus_config("WAVETABLE", "aux 0-1 out")
bus_config("REVERBIT", "aux 0-1 in", "out 0-1")
/* A WAVETABLE hurok és a REVERBIT hívás hosszát vezérli. */
totdur = 6
reset(2000) /* enélkül a WAVETABLE rövid hangjai kattognának */
/*-----*/
setline(0,0, 1,1, 5,0)
makegen(2, 10, 10000, 1,.5,.3,.1)
dur = 1
amp = 10000
freq = 9.00
pctleft = 1
srand(3284)
dur = .08
for (st = 0; st < totdur; st = st + .3)
    WAVETABLE(st, dur, amp, freq, random())
freq = 7.10
for (st = 0; st < totdur; st = st + .45)
    WAVETABLE(st, dur, amp, freq, random())
/*-----*/
amp = 1
revtime = 1.5
revpct = .2
rtchandel = .15
cf = 200
setline(0,1, 1,1) /* felülbírálja a WAVETABLE burkológörbét */
/* az inskip értéke 0 KELL legyen, ha az aux buszról olvasunk */
REVERBIT(st=0, insk=0, totdur, amp, revtime, revpct, rtchandel, cf)
```

képzés paramétereinek valós idejű változtatása, különböző felhasználói felületek létrehozása, valamint egy megbízhatóbb csatoló támogatás megvalósítása.

A CVS (Concurrent Versions System) kiépítésével számos új együtt-

működés és fejlesztés jött létre. Az RTcmix magját a GPL (GNU Public License) alatt jelentettük meg, az egyes összetevők, például a hangszerek és hatások között azonban más feltételekkel terjeszthető elemeket is találhatunk.

Az RTcmix beszerzése és telepítése

Az RTcmix valósidejű képességeinek kihasználásához az OSS linux-os meghajtóit kell telepítenünk, valamint beállítanunk (lásd még a További információk című részben). Az ALSA (Advance Linux Sound Architecture) támogatást nem próbáltuk ki ugyan, de szerepel terveink között. Ha az RTcmixet valósidejű hangfeldolgozásra szeretnénk használni, akkor a hangkártyánknak teljesen duplexnek kell lennie (ez azt jelenti, hogy a kártya egyszerre képes felvenni és lejátszani is). A cikk megszületésekor a legújabb változat száma még a 3.0.2 volt. Az egyes összetevőket külön csomagokban találjuk. Az RTcmix-3.0.2.tar.gz a magot és az ütemezőt, az insts.jg-3.0.2.tar.gz a John Gibson által írt hangszereket, az insts.mch-3.0.2.tar.gz a többcsatornás hangszereket, az insts.std-3.0.2.tar.gz pedig a hagyományos hangszereket tartalmazza.

Térjünk ki most röviden a program szóhasználatára: az RTcmix hangszer lehet hanghatás is (például egy DELAY), de hangképző is (például egy FM szintetizátor).

A hangszercsomagokat a maggal azonos könyvtárba csomagoljuk ki. Ha tehát a fő kódot a /usr/local/src-be helyezzük, akkor végeredményül a /usr/local/src/RTcmix-3.0.0-t kapjuk. Az insts.std.tgz pedig a /usr/local/src/RTcmix-3.0.0/ insts.std-3.0.0-ba fog kerülni. A hangszerek és az RTcmix-változatok következetességéért minden csomagnévben elhelyeztük annak változatszámát is. A fordítást egyszerűbbé tehetjük, ha a hosszú nevek helyett hivatkozásokat használunk, például:

```
ln -s insts.std-3.0.0. insts.std
```

Miután mindent kicsomagoltunk és a megfelelő hivatkozásokat is létrehoztuk, kezdődhet a fordítás. A beállításokat saját magunknak kell elvégeznünk, az RTcmix gyökérkönyvtárban lévő makefile.conf fájl módosításával. Csupán néhány sort kell megváltoztatnunk:

```
# Az RTcmix gyökérkönyvtára:
TOPDIR = /usr/local/src
```

```
# Az RTcmix könyvtárának neve:
CMIXDIR = $(TOPDIR)/RTcmix-3.0.0
```

```
# A hangszerekre mutató hivatkozásokat
# tartalmazó könyvtár:
LIBDESTDIR = $(CMIXDIR)/shlib
```

```
# Itt adhatunk a programhoz új hangszereket:
INST_DIRS = insts.base #insts.std insts.jg
# insts.mch
```

A két könyvtárnév, amit szinte biztos, hogy meg kell változtatnunk: a TOPDIR és az INST_DIRS, attól függően, hogy milyen hangszercsomagjaink vannak. A megfelelő változtatások elvégzése után az RTcmix könyvtárban adjuk ki a make, majd a make install parancsot. Ez mindent lefordít és egy CMIX nevű futtatható fájl helyez el a TOPDIR/bin

könyvtárban, az osztott könyvtárakat pedig a TOPDIR/shlib-be rakja. Győződjünk meg arról, hogy a bin könyvtár a PATH része.

Zeneszerkesztés

A MINC parancsnyelvvvel egyszerűen és hatékonyan vezérelhetjük az RTcmixet. A MINC-parancsfájlokat a program kottafájlként kezeli. Ha a C nyelvet alapjaiban ismerjük, a MINC elsajátítása sem lesz túl nehéz. A két legfőbb különbség, hogy a sorok végére nem kell pontosvesszőt tennünk, és függvények sincsenek. A MINC és az RTcmix

2. lista RT_REVERBIT.sco

```
/*RT_REVERBIT.sco */
/* Valósidejű visszhang a mikrofonbemenethez. */
set_option("full_duplex_on")
rtsetparams(44100, 2, 512)
load("REVERBIT")
rtinput("AUDIO", "MIC")
bus_config("REVERBIT", "in 0", "out 0-1")
dur = 60
amp = .9
revtime = 0.4
revpct = .8
rtchandel = .02
cf = 1000
setline(0,0, 1,1, 9,1, 10,0)
REVERBIT(st=0, insk=0, dur, amp, revtime, revpct, rtchandel, cf)
```

segítségével történő zeneszerkesztés többéves főiskolai tananyag, tehát kizárt, hogy egy cikk keretein belül elmagyarázzam. Ehelyett nézzünk végig néhányat a lehetőségeit bemutató kottafájlok közül. A kottafájl futtatásához küldjük el azt a CMIX-nek egy átirányítás segítségével:

```
CMIX < kottafájl
```

Ekkor a fájl az értelmezőbe kerül, ez az ütemező felé továbbítja. Valósidejű felhasználás esetén az RTcmix értelmezője egy TCP-csatolót figyel, s a bejövő adatokat értelmezi. Ebben a változatban az értelmező és az ütemező a linuxthreads könyvtárat használja. A kottafájlokat John Gibson készítette, és az RTcmix 3.0 csomagban is megtaláljuk ezeket. Az 1. lista egy viszonylag egyszerű, két RTcmix hangszert használó kottafájl mutat be. Természetesen rengeteg ilyen példafájllal találkozhatunk a csomagban. A WAVETABLE egy hullámtábla-kereső, a REVERBIT pedig visszhangosítja a hangfolyamot. Ebben a példában a WAVETABLE kimenete a REVERBIT-hez kerül, majd innen a hangkártyához, illetve egy fájlba. Nézzük végig ezt a kottafájlt:

```
rsetparams(44100, 2)
```

E parancsnak minden RTcmix kottafájlban szerepelnie kell. Hatására a hangkártya sztereó, 44 100 Hz-es üzemmódba vált. Az elhagyható harmadik értékkel a gyorstár méretét állíthatjuk be, ezzel a késést szabályozhatjuk. Az alapértelmezett érték 8192 hangminta. Ezután a használni kívánt „hangszereket” kell megadnunk:

```
load("WAVETABLE")
load("REVERBIT")
```

Ahogy már említettük, az RTcmix támogatja a könyvtárak dinamikus betöltését. Ezek a parancsok futásidőben töltik be a hangszereket, így

azok kevesebb memóriát foglalnak el, mint ha mindegyiket egyszerre betöltenénk. A módszer segítségével több hangszert alkalmazó kottafájlokat is készíthetünk.

Az alábbi sorok megjegyzésben láthatók, tehát nem működnek. A jelek kitörlésével a hangot a kártyára vagy hangfájlba küldhetjük. A `set_option` parancs ki- és bekapcsolja az RTcmix bizonyos tulajdonságait. Például a `set_option("AUDIO_OFF")` hatására a hangkártyát nem használja a program; ez olyan esetben hasznos, ha a kottafájl túlságosan leterhelné a gépet. Ha a hangot az egyik helyről a másikra (például a hangkártyára vagy egy fájlba) szeretnénk irányítani, a

```
bus_config("WAVETABLE", "aux 0-1 out")
bus_config("REVERBIT", "aux 0-1 in", "out 0-1")
```

parancsokat használhatjuk.

Már említettem, hogy az RTcmix egyik legújabb szolgáltatása, hogy segítségével megváltoztathatjuk a hangfolyam útját. Hogy e bővítés összhangban maradjon a már jó ideje használt keverési eljárásokkal, bevezettük a „buszokat” (bus). Ezeknek a belső útvonalaknak „aux” a neve; olvasáskor bemenetként („in”), íráskor pedig kimenetként („out”) működnek. A buszok meghatározása a következőképpen történhet: `bus_type "szám" [in vagy out]`. Tehát az "out 0" egy kimeneti útvonal lesz (például egy hangszóró), az "in 0" pedig egy vonal- vagy mikrofonbemenet. Ezeket ne tévesz- szük össze az "aux 0 in" formával, ez ugyanis egy hangszer által olvasott belső busz meghatározása.

Tehát a fenti parancsok először a WAVETABLE kimenetét terelik a 0. és 1. számú auxra. Ezután a REVERBIT `bus_config`-ját úgy állítják be, hogy a fenti két kimenetet olvassa be, s a kimeneteket egy sztereó kártyára küldi (out 0-1). Egy hangszerhez több buszbeállítást is használhatunk, mindegyik a legutóbbi beállítást veszi alapul.

A fájl következő sora a `totdur` globális változót állítja be. A változó a fájl beolvasásakor kapják meg értéküket. Foglalt változók nincsenek, tehát bármilyen nevet használhatunk.

A hangképzés belső értékei (például az amplitúdógörbe) frissítéséhez a `reset` parancsot használjuk. A `reset (2000)` a másodpercenkénti 44 100 hangmintát 2000-szer frissíti, tehát átlagosan minden 20. minta frissül.

A következő két sor már titokzatosabb:

```
setline(0,0, 1,1, 5,0)
makegen(2, 10, 10000, 1,.5,.3,.1)
```

Az RTcmix, hasonlóan a hosszú évek alatt kifejlesztett programokhoz, a régebbi rendszerekkel való együttműködés megőrzéséért rengeteg régi szolgáltatást is tartalmaz. A generátorokat a `makegen` paranccsal határozhatjuk meg. Ezekkel a hangszer számos tulajdonságát megváltoztathatjuk: az amplitúdógörbét, a hullámformát, a vibratót stb.

A `setline` parancs is ide kapcsolódik: segítségével itt az amplitúdógörbét adjuk meg. Az idő-érték párokkal a 0. időpontban 0-t, az 1. időpontban 1-et, az 5.-ben pedig 0-t állítunk be. Az idő itt viszonylagos, tehát a görbe időtartama a hang („note”, lásd lejjebb) hosszától függ. A WAVETABLE hangszernek szüksége van valamilyen hullámformára a hang létrehozásához. Ezt az adatot a 2-es tárolóba (slot) helyezi, míg az 1-es tárolót a `setline` által létrehozott amplitúdógörbe tárolására tartja fenn. Ennek a `setline` parancsnak a helyére egy `makegen` parancsot is írhatnánk (gen24). A `gen10` a megfelelő tárolóba (most a 2-esbe) egy szinuszhullámot tölt be. A hullám tábla mérete 10 000 hangminta. A többi paraméter a harmóniasor következő elemeinek amplitúdóját határozza meg. Ennél bonyolultabb már nem lesz a dolog, már ami ezt a cikket illeti...

Néhány további globális változó meghatározása után alaphelyzetbe

Kapcsolódó címek

Az RTcmix linuxos változatának címe:

➔ <ftp://presto.music.virginia.edu/pub/Rtcmix>

A Virginia Center for Computer Music, az RTcmix fejlesztésének központja:

➔ <http://www.virginia.edu/~music/VaCenterCompMusic.html>

A virginiai egyetemen működő McIntire Department of Music tanszéken zeneszerzést tanulhatunk. A legújabb PhD programjuk kiváló lehetőséget jelent a számítógépes zenében komolyabban elmélyülni kívánók számára:

➔ <http://www.virginia.edu/~music>

A Princeton Sound Kitchen honlapján a MINC parancsokról és a Cmix felépítéséről olvashatunk:

➔ <http://www.music.princeton.edu/winham/cmix.html>

A Columbia Egyetem Computer Music Center részlegének honlapján rengeteg érdekes leírást és kottafájlt találhatunk:

➔ <http://www.music.columbia.edu/cmix/>

John Gibson honlapja:

➔ <http://www.people.virginia.edu/~jgg9c/musi540/index.html>

Példafájlokat és az RTcmixszel kapcsolatos adatokat találhatunk a következő címen is:

➔ <http://presto.music.virginia.edu/RTcmix/linuxjournal/>

A cincinnati egyetem honlapjáról különböző RTcmix kezelői felületeket tölthetünk le:

➔ <http://meowing.ccm.uc.edu/>

OSS:

➔ <http://www.opensound.com/>

Az RME Digi96 sorozatú hangkártyák:

➔ <http://www.rme-audio.com/>

Az ICMC és a SEAMUS honlapjai:

➔ <http://www.computermusic.org>

➔ <http://seamus.lsu.edu/>

állítjuk a véletlenszám-generátort a `srand(3284)` paranccsal.

A zenét a következő sorok hozzák létre:

```
for (st = 0; st < totdur; st = st + .45)
WAVETABLE(st, dur, amp, freq, random())
```

Ez egy egyszerű hurok, amely addig szól, amíg az `st` nagyobb, mint a korábban meghatározott `totdur`. A WAVETABLE paramétereire (csakúgy mint az RTcmix más szolgáltatásaira és hangszereire) p-mezőkkel hivatkozunk. Ezek a WAVETABLE esetében a következők:

- p0 = indulási idő
- p1 = hossz
- p2 = digitális amplitúdó (itt: 0 és 32 767 között vehet fel értéket)
- p3 = frekvencia (Hz)
- p4 = pan

Ha p4=0, akkor a teljes kimenet a fentebb található `bus_config` által meghatározott 0-ás csatornára kerül; ha p4=1, akkor pedig az 1-esre. 0,5 megadáásával a kimenet egyenlő mértékben oszlik el a csatornák között. Figyeljünk arra, hogy a `random()` egy 0 és 1 közötti

lebegőpontos értékkel tér vissza.

A kotta utolsó néhány sora az amplitúdógörbét a `setline(0,1,1,1)` paranccsal alaphelyzetbe állítja, mert a REVERBIT-hez egy kissé különböző görbére lesz szükség. Ezt a REVERBIT (`st=0, insk=0, totdur, amp, revtime, revpct, trchannel, cf`) segítségével hívhatjuk meg.

A REVERBIT p-mezői a következők:

- p0 = indulási idő
- p1 = az indulástól a bemenet olvasásának megkezdéséig eltelt idő (A `bus_config`-ok esetében 0-t használjunk, fájlból olvasáshoz pedig valamilyen más értéket.)
- p2 = hossz
- p3 = amplitúdószorzó (általában 0 és 1 közötti szám)
- p4 = a visszhang hossza
- p5 = a visszhang százalékos aránya
- p6 = a jobb oldali csatorna késleltetése (delay)
- p7 = az aluláteresztő szűrő levágási frekvenciája (Hz)
- p8 = DC blokkoló szűrő hozzárendelése (logikai érték, elhagyható).

Ennyi volna. A kotta futtatásához írjuk be az alábbi parancsot:

```
CMIX < WAVETABLE_REVERBIT_1.sco
```

A 2. lista zeneileg egyszerűbb példa, mégis egy rendkívül hasznos tulajdonságot mutat be. Segítségével a bejövő hangfolyamot valós időben dolgozhatjuk fel. A kétirányú adatforgalomra a kártyát a

```
set_option("full_duplex_on")
```

paranccsal készíthetjük fel, ezt mindenképpen az `rtsetparams` előtt kell lefuttatnunk, hiszen az néhány, az `rtsetparams` által használt jelzőt is beállít. Az előző példához hasonlóan 44 kHz-es sztereo üzemmódba lépünk, de a belső gyorstár méretét 512 hangmintára csökkentjük, a valósidejű adatfeldolgozás gyorsítása céljából.

Az `RTcmix` az `rtinput("AUDIO", "MIC")` parancs hatására a hangbemenetekről olvas. A `MIC` paraméter csak az IRIX-változatban szükséges. Hangfájl beolvasásához egyszerűen az `rtinput("fájlnev.aiff")` parancsot kell használnunk.

Összegzés

Az `RTcmix` egy rendkívül sokoldalú és könnyen használható, nyílt forráskódú programcsomag. A jelenlegi szolgáltatások számos programozó munkáját dicsérik, s a fejlődés még ma is tart. A Linux tökéletes platformként szolgál a fejlesztéséhez és a használatához egyaránt. A csúcsmínőségű hangkártyák linuxos támogatásának megjelenésével az `RTcmix` képességei az operációs rendszerrel együtt bővülnek. Itt a rengeteg szolgáltatásnak csupán töredékét mutathattam be, a sorozat további részeiben azonban a hangszerek készítésével és valósidejű vezérlésükkel is foglalkozunk. Izgalmas, hogy egy számítógép, egy operációs rendszer és egy programcsomag milyen hasznos segítőtársa lehet egy zenésznek.

David Topper (topper@virginia.edu)

a virginiai egyetem zenei központjának munkatársa.

A Linux azóta kedvenc operációs rendszere, amióta letöltötte a 40 lemeznyi Slackware-t, az 1.0.9-es rendszerrel együtt. Makacsul bízik abban, hogy a Linuxhoz hasonló nyílt forrású rendszerekkel a számítógép olyan fontos eszközzé válhat korunk emberének, mint a távcső az ókori csillagászoknak. Honlapja az alábbi címen érhető el:

☞ <http://www.people.virginia.edu/~djt7p>

