

Alkalmazásaink kipróbálása az Xnee segítségével

Elfáradt a kattintóujjunk programunk grafikus felületének próbálgatásakor? Akkor foglaljuk héjba gonosz tesztjeinket az Xnee-vel!

Az Xnee a felhasználói tevékenység rögzítésére és ezek újrátjátszására képes egy programon belül. Ha a program használatát rögzítjük vele, akkor az Xnee-vel később is kipróbálhatjuk azt. Ezeket a próbákat minden változat kiadása előtt (vagy akár minden este) elvégezhetjük programunk minőségének megőrzése érdekében. Valóban ilyen egyszerű lenne az egész? Majdnem.

Az Xnee nem csak grafikus felületek kipróbálására szolgál: a programot parancssorból indítható programjaink kipróbálására is használhatjuk. Ehhez néhány próbafájl kell létrehozni, amelyek a parancssori alkalmazás összes kapcsolójának a működését ellenőrzik, majd a végén elemezhetjük az eredményeket. Az Xnee-t már annak vizsgálatára is használták, hogy mekkora adatforgalom bonyolódik le egy olyan nagyobb méretű hálózaton, amely számos, kis sávzélességgel rendelkező ügyfélgépből áll. Ha ugyanazokat a próbákat egyszerre több gépen is el szeretnénk végezni, akkor a program később kifejlesztett többgépes szolgáltatását használhatjuk. Az Xnee a programok kipróbálásán kívül azok bemutatására is alkalmas – ebben az esetben türelmes előadóként, fáradtság nélkül végzi a munkáját.

Az Xnee története

1997-ben *Henric Johansson* és jómagam PhD-dolgozatunk témájaként az X-események rögzítését és újrátjátszását választottuk. Egy svéd vállalatnak készítettünk egy – nem ingyenes – felvevőt és újrátjátszót, amit belső használatra kértek. Miután munkát találtam, szükségem volt egy ingyenes próbaprogramra X11 alá, tehát úgy döntöttem, hogy a dolgozatom körüli munka során összegyűlt tapasztalataimból merítve írok egyet. Az Xnee fejlesztése 1999 nyarán kezdődött és az elejétől fogva a GPL szerződés vonatkozott rá. 2002 novemberében jelent meg az 1.0-s változat és 2003 februárjára az Xnee GNU-csomag lett.

Bemutató

Mielőtt belekezdenék az Xnee ismertetésébe, az alábbi rövid bevezetésben elmagyarázom a cikkben szereplő kifejezéseket. Az X egy grafikus-ablakos felhasználói felület többféle felületre; az X-kiszolgáló egy program,

amely az eszközök kezelését és a képernyőre rajzolást végzi. A GNU/Linux és rendszereken az Xfree86 a leggyakrabban használt X-kiszolgáló. Az X-programokra ügyfél néven hivatkozunk, ilyen például az `xterm` vagy a Galeon. Az ügyfelek az X-protokoll segítségével érik el az X-kiszolgálót.

Ebben a cikkben az X-kiszolgáló és az ügyfelei közti adatcserére szolgáló csomagküldést vesszük górcső alá. A csomagok nevei az alábbiak lehetnek: `Event` (esemény), `Request` (kérelem), `Reply` (válasz) és `Error` (hiba) – ezekre itt protokolladatként hivatkozunk. Az alábbi lista az X11-protokolladatok leírását tartalmazza:

- Az ügyfél kérelmet küld a kiszolgálóhoz, ami egy bizonyos feladat elvégzésére vagy adatok lekérésére vonatkozik.
- A kiszolgáló választ küld az ügyfélnek. Nem minden kérelmére érkezik válasz.
- A kiszolgáló az `Event` csomagot elküldi az ügyfélnek, ezzel értesíti arról, hogy felhasználói adatbevitel vagy valami olyasmi történt, amiről az ügyfélnek tudnia kell (például már nem az ő ablaka aktív).
- Ha a kérelem érvénytelen volt, a kiszolgáló az `Error` csomagot küldi el az ügyfélnek.

A legérdekesebb, hogy minden olyan esetben, amikor a felhasználó az egér vagy a billentyűzet segítségével kapcsolatba lép a számítógéppel, az X-kiszolgáló egy vagy több eseménycsomagot küld el az ügyfélnek. Ezek némelyike közvetlenül a felhasználói tevékenység hatására jön létre – az ilyen eseményeket eszközeseményeknek nevezzük. Ezek az alábbiak lehetnek: `ButtonPress` (az egérgomb lenyomása), `ButtonRelease` (az egérgomb elengedése), `MotionNotify` (az egérmutatató mozgása), `KeyPress` (a billentyű lenyomása) és `KeyRelease` (a billentyű elengedése). Ha tehát mindezen eseményeket sikerülne rögzítenünk, akkor a felhasználói tevékenység minden részletét tárolhatnánk. Ha lenne egy olyan robotunk, amelyik képes lenne a tárolt adatokat értelmezni, majd az egérrel és a billentyűzettel újra lejátszaná az eseményeket, máris lenne egy próbarobotunk. Vagy ami még ennél is jobb: ha megoldhatnánk ezeknek az eseményeknek a „hamisítását” (programból történő előállítását), akkor programból megvalósított próbarobotról beszélhetnénk. Szerencsé-

re az X az események rögzítését és újrátjátszását is támogatja. Az X-protokollon átvitt adatok rögzítéséhez a RECORD vagy az XTrap bővítményeket (plug-in) kell használnunk. A rögzítést más módon is megoldhatjuk, például az X-foglalat (socket) figyelésével, de most a RECORD-dal, az Xnee által is alkalmazott módszerrel foglalkozunk. A visszajátszás alatt a RECORD bővítmény hangolja össze a rögzített adatokat. A RECORD az ügyfelek és a kiszolgáló között mozgó adatok másolatát elküldi az őket kérő ügyfélhez. A RECORD bővítménnyel tehát az Xnee az összes rögzíteni kívánt protokolladat fájlban való tárolására képes, hogy azután egy későbbi időpontban újrátjátszhassa őket. Az Xtest bővítmény az összes eszközeseményt hamisítani tudja, segítségével az Xnee egy élő felhasználó tevékenységét utánozhatja, például az egér mozgását, a billentyűk vagy valamelyik egérgomb lenyomását és elengedését – más adat nem állítható elő vele.

Az Xnee telepítése

Az Xnee egy GNU-csomag, forráskódját az Xnee honlapján találjuk. Töltsük le a legfrissebb forrást, amely cikkünk írásakor az 1.0.6-os változat! Bontsuk ki, állítsuk be, építsük fel és telepítsük:

```
tar zxvf xnee-1.0.6.tar.gz
cd xnee-1.0.6
./configure
make
make install
```

A honlapról RPM csomagokat is letölthetünk, s az Xnee a FreeBSD ports tree-ben is elérhető. Az Xnee felhasználói és fejlesztői leírással érkezik, amelyek többféle formátumban is megtalálhatók a csomagban. A TeX formátumú kézikönyvek a programmal együtt érkeznek és a GNU FDL vonatkozik rájuk. Ha nem mi magunk szeretnénk a dokumentumokat felépíteni, akkor töltsük le őket valamilyen más formátumban (PDF, HTML, INFO, TXT) az Xnee honlapjáról. Jelen pillanatban az Xnee leírása az 1.0.4 változatnál jár. Bontsuk ki a fájlokat:

```
tar zxvf xnee-doc-1.0.4.tar.gz
```

Az Xnee futtatásakor ellenőrizzük, hogy a RECORD bővítmény engedélyezve van-e. XFree86 alatt ehhez a RECORD modult kell betöltenünk. Nyissuk meg az XFree86 beállítófájlt (ami általában a */etc/X11/XF86Config-4*), és nézzünk bele a `Module` részbe, ahol az alábbi sornak kell szerepelnie:

```
Load "record"
```

Erről az Xnee GYK-ban találunk további útmutatást.

Egyszerű példák a program használatára

Nem mélyedünk el azonnal az Xnee részletes taglalásában, előbb egy egyszerű példával kezdünk. Indítsuk el az Xnee-t a `--all-events` kapcsolóval, ami néhány esemény rögzítését állítja be. Ha komolyabban akarjuk használni a programot, akkor ez a kapcsoló kerüendő, de ebben a példában megfelelő lesz:

```
xnee --all-events
```

Amikor az egérmutatót vagy a billentyűket mozgatjuk, esetleg egérgombokat nyomunk meg, az Xnee adatokat ír ki az eseményről. Most egy egyszerű eseménysort rögzítünk, amelyet azonnal vissza is játszunk. Húsz egérmousegészítéshöz így indítsuk el az Xnee-t:

```
xnee --record --out session1.xnr
--device-event-range MotionNotify --loops 20
```

A parancssori kapcsolók és paraméterek jelentése: a rögzítés bekapcsolása (`--record`), az eredmények mentése a *session1.xnr* fájlba (`--out session1.xnr`), a `MotionNotify` események rögzítése (`--device-event-range MotionNotify`) és összesen húsz esemény tárolása (`--loops 20`). Az eseménysor visszajátszásához az alábbi formát használjuk:

```
xnee --replay --file session1.xnr
```

A parancssori kapcsolók jelentése: a visszajátszás üzemmódot használjuk (`--replay`) és a *session1.xnr* fájlból vesszük az adatokat (`--file session1.xnr`).

A rögzítés határainak beállítása

Az Xnee-nek a határok beállításával magyarázhatjuk el, hogy mit kell rögzíteni. A határok kezdő (start) és záró (stop) értékkel rendelkeznek. Az alábbi adatokat rögzíthetjük: magkérelmek (`core-request`), eszközesemények (`device-event`), átvitt események (`delivered-event`), hibák (`error`), válaszok (`reply`), bővítési kérelmek (`extension requests`) és bővítési válaszok (`extension reply`). A cikkben nem elemezzük őket, aki bővebben szeretne erről a témáról olvasni, az nézze meg a RECORD bővítménydokumentumokat.

Lássunk azért néhány példát! A `MotionNotify` eszközesemény rögzítéséhez az alábbi kapcsolót használjuk:

```
--device-event-range MotionNotify
```

Ha a `KeyPress`-től a `MotionNotify` és `CreateNotify` eseményekig terjedő tartományt kívánjuk rögzíteni, akkor a következő kapcsoló használatára van szükségünk:

```
--device-event-range KeyPress, MotionNotify,
--CreateNotify
```

Ha rövidebb parancssort szeretnénk, akkor az eseménynévhez tartozó számot is használhatjuk a nevek helyett. A rögzíteni kívánt adat számát az Xnee `--print-data-name` kapcsolójával kereshetjük meg:

```
xnee --print-data-name
```

Az Xnee leállítása

A munkát a rögzíteni kívánt adat számának beállításával (`--loops` kapcsoló) állíthatjuk le, vagy a `TERM` jel elküldésével (nyomjuk le a `CTRL+C` billentyűket abban a terminálablakban, ahonnan az Xnee-t indítottuk) megszakíthatjuk a program futását. Azt is megtehetjük, hogy egy billentyűt-

1. táblázat X-események a Galeon-próba kezdetekor

A protokolladat neve	A felhasználó/ügyfél tevékenysége
MotionNotify	A felhasználó az egérmutatót a Galeon indítóikonjára viszi.
ButtonPress	A felhasználó megnyomja az egérgombot és a Galeon elindul.
CreateNotify	A Galeon elindul és az ablak megnyílik.
VisibilityNotify	A kezdőlap betöltődik és a felhasználó számára láthatóvá válik.
MotionNotify	A felhasználó a betöltött oldalon egy hivatkozás fölé viszi az egérmutatót.
ButtonPress	A felhasználó a hivatkozásra kattint.
VisibilityNotify	Az új oldal betöltődik és a felhasználó számára láthatóvá válik.

kapcsolatot hozunk létre, amelyet rögzítés közben semmi másra nem fogunk használni. Ehhez a `--stop-key` kapcsoló szükséges. Ha azt szeretnénk, hogy az Xnee a `CTRL+ALT+A` billentyűk lenyomására hagyja abba a rögzítést, a parancsot egészítsük ki az alábbiakkal:

```
--stop-key Control+Alt,a
```

Összehangolás

Vajon miért foglalkoznánk az eszközeseményeken kívül más adatok rögzítésével, ha úgysem tudjuk visszajátítani őket? Az Xnee ezeket az adatokat az összehangolásához használja, és a dolgok itt kezdenek egy kissé bonyolulttá válni.

Tegyük fel, hogy a Galeon vagy más webböngésző használatát szeretnénk rögzíteni. Rögzítés közben minden nagy-szerűen működik és a hálózat is kifogástalanul használható, amikor azonban visszajátsszuk a Galeonban történeteket, nem tudjuk elérni az internetet. Ha a Galeon nem képes betölteni az oldalt, akkor a visszajátzást érdemes addig folyamatban tartani, amíg a hálózat újra elérhetővé nem válik és az oldal betölthető nem lesz.

A többi adat rögzítésével a programmal való munkát hangolhatjuk össze. Ha például a weboldalnak a Galeon ablakában megjelenített adatokat rögzítjük, akkor visszajátzaskor ugyanezeknek az adatoknak a továbbítására kell várnunk. Ez biztosítja, hogy az oldal betöltődik, mielőtt az események visszajátzása elkezdődne. Példánkban az egyszerűség kedvéért sok X-protokolladatot kihagytunk (1. táblázat). Az egyszerű eseményhalmaz visszajátzásokor az Xnee ugyanezeket az eseményeket használja (2. táblázat).

Mit hangoljunk össze mivel?

Bár az összehangolásra szükség van, nehéz feladat a hozzá használni kívánt adatok megtalálása. Az Xnee ezt bővítményekkel oldja meg, amelyek azt határozzák meg, hogy bizonyos alkalmazásoknál mely eseményeket kell rögzíteni. Ezeknek a bővítményeknek a neve a kipróbálni kívánt alkalmazás nevére utal. Ha egy általunk írt webböngészőt szeretnénk kipróbálni, akkor jó ötlet a

2. táblázat Visszajátzás

A protokolladat neve	Az Xnee által végrehajtott művelet
MotionNotify	Az Xnee az egérmutatót a Galeon indítóikonjára viszi.
ButtonPress	Az Xnee megnyomja a gombot és a Galeon elindul.
CreateNotify	Az Xnee ennek az eseménynek a továbbítására vár. Amikor a <code>CreateNotify</code> eseményt megkapja, a fájlban tárolt soron következő eseményre tér rá.
VisibilityNotify	Az Xnee ennek az eseménynek a továbbítására vár. Mivel a hálózat nem él és az oldal nem tölthető be, ez az esemény nem kerül továbbításra. Az Xnee tovább vár, majd végül az esemény továbbítódik és az Xnee folytathatja a visszajátzást.
MotionNotify	Az Xnee a betöltött oldal egyik hivatkozása fölé állítja az egérmutatót.
ButtonPress	Az Xnee a hivatkozásra kattint.
VisibilityNotify	Az új oldal betöltődik és láthatóvá válik.

Galeon bővítmény használata. Néha viszont nem találunk a programunkhoz megfelelő bővítményt, így ránk hárul a feladat, hogy az összehangolni kívánt protokolladatok megtaláljuk. A következő példa remélhetőleg könnyít a helyzeten: most a `numeric` nevű programhoz kellene a megfelelő adatokat megtalálnunk. Először is indítsuk el a `numeric-et`, majd egy terminálablakban az Xnee-t az alábbi kapcsolókkal:

```
xnee --delivered-event-range EnterNotify-
↳ MappingNotify --human-printout --loops 1000
```

Ennek hatására egy csomó érdektelen esemény tölti be a képernyőt, tehát állítsuk le az Xnee-t. A nem kívánt eseményeket a határok beállításával szűrjük ki:

```
xnee --delivered-event-range EnterNotify-
↳ KeymapNotify, VisibilityNotify-MappingNotify
↳ --human-printout --loops -1
```

Így már sokkal jobb. Kezdjük meg a rögzítést az alábbi parancsokkal:

```
xnee --delivered-event-range EnterNotify-
↳ KeymapNotify, VisibilityNotify-CirculateRequest,
↳ SelectionClear-MappingNotify --loops 1000 --out
↳ session1.xnr
```

Írjunk valamit a `numeric` táblázatba és a menükkel illesszük be az aznapi dátumot vagy más adatokat. Ha elkészültünk, váltsunk át a terminálablakra és a `CTRL+C` billentyűvel állítsuk le a rögzítést. Elérkezett az idő az események visszajátzására. A `numeric-et` hagyjuk pontosan olyan állapotban, mint a rögzítéskor

volt, és visszajátszási módban indítsuk el az Xnee-t:

```
xnee --replay -f session1.xnr
```

Az Xnee visszajátszás közben néha megáll egy pillanatra. Ez akkor történik, ha a protokolladatok nem a rögzítés sorrendjében kerülnek visszajátszásra. Ilyenkor az Xnee a futást megállítja addig, amíg a várt adatok (a fájlból beolvasva) a kiszolgálótól meg nem érkeznek. Végül az előre beállított várakozási időtartam leteltekor az Xnee újra próbálkozik. Ha a rögzített adatforgalmat és a visszajátszáskor elküldött adatokat nem tudja összehangolni, akkor bedobja a törölközőt.

Az Xnee-ben a bővítmények segítségével a rögzítés tulajdonságait is beállíthatjuk. Ha megtaláltuk alkalmazásaink beállításait, egy bővítményfájlban mentjük őket. E fájl formája a parancssori kapcsolók használatára emlékeztet. Új bővítményt a legegyszerűbben egy régi másolásával, módosításával és másik néven történő mentésével hozhatunk létre. Az Xnee-csomagban több ügyfélhez is találhatunk bővítményeket. Ha van kedvünk, az általunk készített bővítményeket elküldhetjük az Xnee fejlesztőinek – ehhez az Xnee honlapján találhatunk útmutatásokat.

Ha a programunk új ablakokat nyit meg a felhasználói adatbevitelhez, gondoskodnunk kell arról, hogy ezek ugyanazon a helyen jelenjenek meg. Az Xnee minden eszközesemény koordinátáját programunk kezdőablakához képest határozza meg, nem pedig az éppen megnyitott ablakhoz képest.

A rögzítés megkönnyítése érdekében készítsünk parancsfájlokat, amelyek egy adott feladat szempontjai szerint indítják el az Xnee-t. A parancsfájlhoz programindítót vagy új menüelemet hozhatunk létre az ablakkezelő menüben.

Összegzés

Az Xnee fejlesztése az utóbbi időben látványosan felgyorsult, javarészt a felhasználók hozzájárulásának köszönhetően. Remélem, hogy egyre többen választják az Xnee-t mindennapi társként a programfejlesztésben. Kellemes próbálgatást!

Linux Journal 2004. január, 117. szám



Henrik Sandklef (hesa@gnu.org)

Feleségével és lányával a svédországi Gothenburgban él. A legtöbb idejét a családjával, főzéssel, GNU-programok boncolgatásával és kipróbálásával tölti, emellett alkalmanként a labdarúgással is megpróbálkozik.

KAPCSOLÓDÓ CÍMEK

GNU ➔ <http://www.gnu.org>

Kenton Lee ➔ <http://www.rahul.net/kenton/xsites.html>

XFree86 ➔ <http://www.xfree86.org>

Xnee ➔ <http://www.gnu.org/software/xnee/www/index.html>

