

Növeljük az X hatékonyságát!

Fogjuk munkára az X-et parancsfájlokkal, testreszabott billentyűzettel és párbeszédablakkal!

A Linux rendelkezik grafikus felhasználói felülettel (GUI), amelyen több ablakkezelő és menürendszer is található. De a sebességet tekintve a grafikus felület szűk keresztmetszetet jelent az ember és a számítógép kapcsolatában: a gép sebességétől függetlenül ugyanannyi időt vesz igénybe egy olyan folyamat, amelyik tíz egérekattintást igényel tíz különböző helyen. A parancssorban a hagyományos Unix válasza a szerszámoszláda-szemlélet. Mindent gépesíthetünk parancsfájlokkal, és apró, egy adott feladatot végrehajtó programokat csövezetékkel köthetünk össze. A harmadik megoldás ezt a két megközelítést egyesíti, és elég egyszerű és hatékony ahhoz, hogy a legtöbb célra megfeleljen. Ablakkezelőtől függetlenül bármilyen X-környezet a szokásosnál gyorsabbá és rugalmasabbá tehető.

Saját egér és billentyűzet

Több ujjunk van, mint kezünk, és 101 billentyű jut egy egérre. Bár a rajzolás és az ehhez hasonló feladatok gyorsabban elvégezhetők egérrel, az ujjainknak eközben is a billentyűzeten kell lenniük. Ha egy parancs vagy választási lehetőség egyetlen művelettel elvégezhető, egy-két billentyű lenyomása sokkal gyorsabb az egérekattintásnál, nem beszélve az ínhüvelygyulladás veszélyeiről. Az is jó lenne, hogy amikor az egérhez érünk, azonnal azt tegye, amire szükségünk van. Az `xmodmap` segítségével a billentyűk és az egérgombok kiosztása megváltoztatható, így különleges jelentéssel ruházhatók fel (például ékezetes betűk), vagy GUI-műveleteket rendelhetünk hozzájuk (például az ablak maximalizálását). Az egyik előny, hogy a balkezes emberek a két egérgombot megcserélhetik. Adjuk hozzá a következő sort a `.xinitrc` állományhoz, vagy tegyük az idézőjeles részt a `.xmodmaprc` állományba:

```
xmodmap -e "pointer = 3 2 1"
```

Az utóbbi időben az `xmodmap` már kezelni tudja az egérgörgőket Linux alatt (lásd a *Kapcsolódó címek* részt). Gyorsabbá is tehetjük a görgőkkel való munkát, ha a leggyakoribb műveleteket az újfajta egerek különleges gombjaihoz rendeljük. Például az IntelliMouse Explorerhez ez a bűvös sor szükséges:

```
xmodmap -e "pointer = 1 2 3 6 7 4 5"
```

Amikor a billentyűzetről beszélünk, a módosító olyan billentyűt jelent, ami megváltoztatja más billentyűk állapotát vagy hatását. Szabványos módosító a SHIFT, CTRL, LOCK, ALT és öt további, amelyeket egyszerűen MODN-nek (N=1,2,..5) hívnak. Az `xmodmap` tetszőleges billentyűhöz rendelheti hozzá őket. A leggyakoribb talán a sűgőoldalon is leírt eset, amikor a CTRL és a CAPS LOCK billentyűt cserélik fel. Ugyanígy a Windows billentyű mod4-re változtatható – feltéve, hogy a billentyűkódja 115:

```
xmodmap -e `keycode 115 = Alt_R Meta_R`
```

Grafikus előtétprogram a szolgáltató választásához

```
#!/bin/bash

/bin/rm -f /tmp/netsettings

echo -n "PROVIDER=" > /tmp/netsettings
Xdialog --menubox "Szolgáltató választása"
    ↵20 40 5
"ISP_1" "Munkaidőben a legolcsóbb"
"ISP_2" "Hétfvégén gazdaságosabb"
"ISP_3" "Leggyorsabb FTP" 2>>
    ↵ /tmp/netsettings

RETVAL=$?
# add control code here

echo -n "ACCOUNT=" >> /tmp/netsettings
Xdialog --inputbox "írd be a fiók nevét"
    ↵10 25 "son" 2>> /tmp/netsettings

echo -n "LOGFILE=" >> /tmp/netsettings
Xdialog -fselect /tmp 20 80 2>>
    ↵ /tmp/netsettings

source /tmp/netsettings
netconn $PROVIDER $ACCOUNT $LOGFILE
```

A billentyűk alkalmazások indítására is programozhatók, de ez már függ az ablakkezelőtől. A Blackbox 0.65-ben például a `bbkeys` segédprogrammal lehet billentyűleüetéseket és alkalmazásokat egymáshoz rendelni. A `$HOME/.bbkeysrc` állományban a

```
KeyToGrab(F1), WithModifier(None),
    ↵WithAction(ExecCommand), DoThis(xterm
    ↵-geometry 80x25 -e mutt)
```

szor azt eredményezi, hogy az F1 billentyű lenyomására egy 80×25 méretű `xterm` ablakban elindul a `mutt`. Számos más ablakkezelő is lehetővé teszi az ilyen hozzárendeléseket – olvassuk el a leírásokat.

Kattintgatós parancsfájlok

Minden parancsfájl használhat ablakokat a felhasználóval való párbeszéd során, és minden grafikus ügyfél (maga az X is) veheti a bemenetét közvetlenül egy szöveges programból. Az első eset akkor következhet be, ha a parancsfájlnak a felhasználó visszajelzésére van szüksége, de ez a hagyományos módon nem lehetséges; például a felhasználó inkább meghal, csak ne kelljen gépelni, vagy nincs billentyűzet (utcai termi-

nálok). A második esetre példa, amikor a felhasználónak előre nem kiszámítható szöveget kell egy konzolos programból egy olyan X-es programnak átadnia, amelybe a szöveg kézzel nem másolható át. Gyakran előfordul ez a mindennapi életben: a `mutt` és a `lynx` elegendő lehet a levelek olvasására és a web-böngészésre, de mi van, ha valamelyik levél vagy weboldal azt mondja: „Nézd meg ezt a filmelőzetest”, és ekkor a Mozillát indítanánk el egy egérgattintással...?

Parancsfájlok bővítése ablakkal

Legalább tíz éve már X-ablakokat is lehet parancsfájlokból nyitni. Kezdetben ott volt a ma már nem működő Xscript. A korszerű megoldás az Xdialog, a GTK+ alapú párbeszéd-ablak-építő. A *listánkban* közölt parancsfájl akár a valós életből is származhatna: a felhasználó kiválaszthatja a legjobb internetszolgáltatót, a használandó fiókot és a kapcsolat naplójának a helyét. Ezután elindul egy hagyományos `pppd/chat` parancsfájl (itt `netconn` a neve), ami átveszi a grafikus felületről begyűjtött adatokat. Vizsgáljuk meg részletesen a grafikus felületet!

A parancsfájl azzal indul, hogy a felhasználó régi választásait tartalmazó `netsettings` állományt eltávolítja. A `netconn` parancsfájl által igényelt összes változó a `bash` parancsértelmező értékadási szabályai szerint a `/tmp/netsettings` állományba kerül. A bal oldali rész egyszerűen soremelés nélkül kiíródik (`echo -n`). A jobb oldali részt, azaz a felhasználó által választott értéket az Xdialog adja át.

Az első párbeszédablakban (1. kép) a felhasználó kiválaszthatja a legjobb internetszolgáltatót. Megadjuk a párbeszédablak leírását (szolgáltató választása), az ablak magasságát (20) és szélességét (40), valamint a beviteli mezők magasságát (5). Az 1. képen bemutatott esetben, miután a felhasználó megnyomta az **OK** gombot, a `/tmp/netsettings` a következő sort fogja tartalmazni:

```
PROVIDER=ISP_1
```

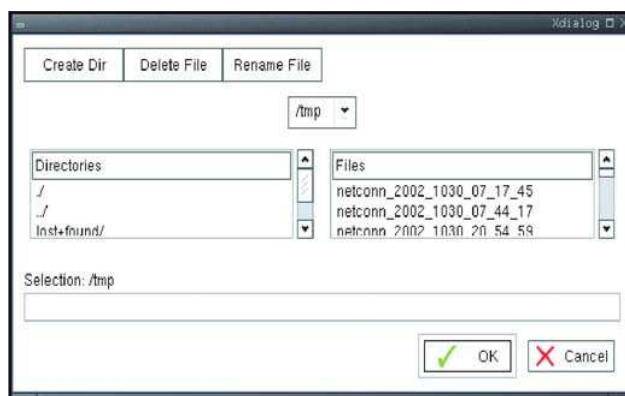
Az Xdialog minden meghívása után a kilépési kód vizsgálatával (a `RETVAL` az 1. listában) érdemes hibaellenőrzést végeznünk. Így ellenőrizhetjük, hogy a felhasználó mit is tett valójában. A rövidség kedvéért ezt a kódrészt most elhagytuk, de közöljük, hogy a `RETVAL 1` értéke azt jelenti, hogy a „Mégsem gombot nyomták meg”, a 255 azt jelenti, hogy a felhasználó bezárta az ablakot, és a 0 jelenti azt, hogy a választás megtörtént. A második Xdialog parancs lehetővé teszi, hogy a felhasználó begépelje a fiók nevét vagy elfogadjá az alapértelmezett beállít



1. kép Az Xdialog indítása



2. kép A fióknév bevitelle



3. kép A naplófájl kiválasztása

tást (2. kép). Az utolsó párbeszédablak (3. kép) egy fájlválasztó ablakot jelenít meg, amelynek segítségével a felhasználó kiválaszthatja a naplófájl helyét és nevét. Ezután a felhasználó választásai már a `/tmp/netsettings` állományban vannak, ezért ezt az állományt csak be kell tölteni (source), és el kell indítani a kapcsolatot.

Az Xdialog nagyszámú más építőelemet is tartalmaz, például választógombokat, csúszkákat és naptárat. Ezek közül kevés használható a felhasználó számára történő visszajelzésre. A szerző a `--msgbox` kapcsolót használja felbukkanó ablak megjelenítésére, amelyben például fel lehet sorolni, hogy fiókok szerint rendezve hány üzenetet töltött le a `fetchmail`. Vannak még folyamatjelzők, adott idő után eltűnő információs ablakok és fájl megjelenítő ablakok. Az Xdialog használható CD-égető, hang- és videolejátszó, valamint biztonsági mentést készítő parancsfájlok létrehozására – a felhasználási lehetőségeknek csak a képzeletünk szabhat határt.

Mi történik akkor, ha a parancsfájlt akkor kell használni, amikor nem fut az X? Semmi gond, karakteres terminálok is meg lehet menüket és ablakokat jeleníteni. A dialog programot kell használnunk, ami az Xdialog karakteres megfelelője.

Szöveg átadása az X-nek

Az URL-es példára visszatérve: gépelés nélkül is át lehet adni egy címet a böngészőnek, ha az `xclip` programot használjuk, amely az egérrel kijelölt szöveget írja ki. Ezt a programot a `gnome-moz-remote` segédeszközzel együtt használjuk, ami vagy egy új Mozilla-példányt indít el, vagy a már futó Mozilla új ablakában megjeleníti a megadott címet. Írjuk be a következő sort egy parancsfájlba:

```
gnome-moz-remote --newwin "'xclip -o'" &&  
↳ /dev/null &
```

Nevezzük el a parancsfájlt `start_browser.sh`-nak, és rendeljük hozzá egy makrógombhoz egy külső alkalmazásokat indítani képes programban. A `mutt`-ban egy ilyen makró például így néz ki:

```
macro pager \cn "!start_browser.sh\n"  
↳ 'open URL'
```

Ezután, ha a `mutt`-ban egy címet látunk, kijelölhetjük az egérrel és megnyomhatjuk a `CTRL-N` billentyűkombinációt. A parancsfájl belsejében az `xclip` kiírja a kijelölt szöveget, és minden úgy zajlik, mintha kézzel indítottunk volna egy

© Kiskapu Kft. Minden jog fenntartva

böngészőt, és a címsorba ezt a szöveget írjuk volna be. Az xclip-nek rokona az xclipboard és az xcutsel. Ez a két program akkor hasznos, ha meg szeretnénk tekinteni a vágólap tartalmát, illetve amikor a kijelölést két olyan alkalmazás között kell átvinni, amelyek ezt nem támogatják. További tudnivalók a megfelelő súgóoldalakon találhatók.

A kattintás utánzása

Hogyan hitethetjük el az X-szel, hogy az egérrel mozgatjuk a mutatót, és kattintunk a gombokkal, miközben csak a billentyűzetet használjuk? Erre való az xbut program. Egy egyszerű állomány segítségével beállíthatjuk, hogy bizonyos billentyűk mozgassák az egérmutatót és utánozzák a kattintást. Még többre képes az xwit, amellyel közvetlenül egy adott pontra helyezhetjük át az egérmutatót, emellett az ablakok áthelyezése és minimalizálása is lehetséges. Az éppen használt xterm minimalizálásához, majd két másodperc után az eredeti méret visszaállításához a következő parancsot használjuk:

```
xwit -iconify; sleep 2; xwit -pop
```

Ez a megoldás a sokáig futó feladatok elvégzésékor hasznos. A sleep helyére írjuk be a sokáig futó parancsot – az ablak eltűnik, majd a munka végeztével újra megjelenik. Végül bemutatjuk a sokoldalú xte eszközt, ami az xautomation csomag része. Az xte -h parancs kiírja a támogatott X-eseményeket. Az alábbi példa áthelyezi az egérmutatót a képernyő bal szélétől számított 320 képpontra, a felső szélétől számított 50 képpontra levő helyre, és kattint egyet az egyes egérgombbal:

```
xte 'mousemove 320 50' 'mousedown 1'
↳ 'mouseup 1'
```

A mousedown és a mouseup események különállóak, így lehet húzni is, nemcsak kattintani.

Jelszókezelő munkaasztal

Egy gyors és hatékony környezetben a felhasználó minden számára engedélyezett folyamatot a lehető legkönnyebben, leggyorsabban és legbiztonságosabban elindíthat (akár a háttérben, akár interaktívan, akár szöveges, akár grafikus programot), minden olyan gépen, amelyhez hozzáférése van. Természetesen a leghelyesebb mindezt az OpenSSH-n használatával megtenni. De mégha RSA és DSA alapú hitelesítést végzünk is – megtakarítva a jelszavak gépenként történő megjegyzését –, minden kapcsolathoz be kell írunk az SSH-jelszót. Szerencsére az ssh-agent képes megjegyezni a titkos kulcsot vagy kulcsokat, és minden hitelesítési feladatot elvégez, amelyhez a titkos kulcs kell. A gyakorlatban ez azt jelenti, hogy ha az X-et az ssh-agent gyermekfolyamataként indítjuk, az ssh-agent programot minden később indított helyi X-ügyfél használhatja a hitelesítésre. Az ablakkezelőt az ssh-agent gyermekeként az .xinitrc vagy az .Xsession állományban úgy indíthatjuk, hogy az ssh-agent után beírjuk az ablakkezelő nevét. Ha sawfish-t használunk, írjunk „ssh-agent sawfish”-t, és minden, ami az X-munkamenet alatt fut, beleértve az SSH programokat is, használni fogja az ssh-agent-et.

A Gnome az ssh-agent programot alpból elindítja. A KDE-hez úgy adhatjuk hozzá, hogy megkeressük a KDE indítóállományát, és úgy járunk el, mint egy ablakkezelő esetében. Az ssh-add parancs az, amelyik ténylegesen megismerteti a kulcsokat az ügynökkel. Ellenőrizhetjük, hogy az ssh-agent

fut-e, és ismeri-e a kulcsunkat:

```
ssh-add -L
```

Ha a számítógépet őrizetlenül hagyjuk, bárki, aki arra jár, kérdés nélkül azonnal hozzáfér mindenhez, amihez mi is. Jelentkezzünk ki, vagy az ssh-add -D parancsral töröljük a kulcsunkat.

Események azonosítása

Elmondtuk, hogyan lehet egyes billentyűk billentyűkódjait más esemény előidézésére átkódolni, de honnan tudhatjuk meg ezeket a kódokat? A megoldás az xev diagnosztikai eszköz, ami egy eseménytesztelő ablakot nyit meg, és minden ablakeseményt kiír ebbe. A szerző gépén a bal oldali Windows-billentyű lenyomása ezt adja (a „keycode” értéke érdekel minket):

```
KeyRelease event, serial 23, synthetic NO,
↳ window 0x1000001, root 0x46, subw 0x0, time
↳ 1108438536, (175,176), root:(627,425), state
↳ 0x40, keycode 115 (keysym 0xffeb, Super_L),
↳ same_screen YES, XLookupString gives 0
↳ characters: ""
```

Végül, de nem utolsó sorban el szeretnénk dicsekedni a grafikus felülettel rendelkező parancsfájllal, és ehhez képernyőképek kellene. Ebben az esetben is elég az X és az ImageMagick, nem kell ennél csicsásabb előtétprogram. A cikkben használt képek mind a következő néhány szabványos parancsral készültek, amelyekhez a megfelelő leírást a súgóoldalakon lehetjük fel:

```
xwd -out temp_image -frame
xwdtopnm temp_image > fig1.pnm
convert fig1.pnm fig1.png
```

Az első parancs a mutatóval kiválasztott ablakról a kerettel együtt mentést készít. A második és harmadik parancs először PNM, azután PNG formátumra alakítja át a képet. Mondanom sem kell, hogy mindhárom parancs könnyen beilleszthető parancsfájlba, ami megkérdezi a felhasználót, hogy miről készüljön a mentés (képernyő vagy ablak), és hová mentse az eredményt.

Összegzés

Számos felhasználó érzi úgy, hogy egy teljes munkaasztali környezet vagy túl sok mindent tartalmaz, ami lelassítja a számítógépet, vagy túl keveset a különleges igények kielégítéséhez. A cikkben leírt eszközök és módszerek segíthetnek ezeknek a felhasználóknak a termelékenység növelésében, és olyan rendszeren is nélkülözhetetlenek, amelyet felváltva használnak a parancssor és a kattintgatás szerelmesei.

Linux Journal 2003. szeptember, 113. szám



Marco Fioretti

Áramkörtervező mérnök, a szabad programok érdeklik a gépesített áramkörtervezés (EDA) és a hatékony munkaasztal kialakítása (jelenleg a RULE projektet vezeti) szempontjából is. Marco a családjával Rómában él.