Dr. Mileff Péter

Unix/linux operációs rendszer üzemeltetése

7. Előadás

Miskolci Egyetem
Általános Informatikai Tanszék

Az X grafikus felület...

A grafikus felület

- Ha egy operációs rendszer versenyképes szeretne lenni,
  - akkor kétség kívül szüksége van egy grafikus kezelői felületre (GUI).
- **Előnye:**
  - GUI felület számos feladatra egyértelműen barátságosabb és könnyebben kezelhető, mint egy szöveges terminál.
- A Unix/Linux rendszerek grafikus felülete nagyon eltér a Windows rendszerekben megismert grafikus felülettől.
  - Maga a rendszer egy nyitott szabványon alapul
- **X Window System** (vagy egyszerűen **X**),
- Hasonlóan sok más Unix-os fejlesztéshez egy akadémiai projekt (MIT – Athena projekt) eredménye.

Az X Window System röviden

- Az X szabványt az **X Consortium** (http://www.x.org) vezeti
- Nagyon sok operációs rendszerre írtak X megvalósítást
  - sok programcsomag látott napvilágot ezen szabvány alapján.
- Miért fontos a szabvány?
  - Ezen programok a szabvány alapján hiba nélkül képesek egymással kommunikálni,
  - ezáltal a grafikus programok kimenete platform-független lett.

- A Linux egyik leginkább elterjedt X megvalósítását az **Xfree86** szervezet (http://www.xfree86.org) készítette.

Az X felépítése

- Az X egy kliens-szerver alapú architektúrájú rendszer
  - ahol a kliens és a szerver tud hálózaton keresztül is kommunikálni,
    - úgynevezett **X protokoll**on keresztül.
- A grafikus képernyőt egy szerverprogram kezeli,
  - minden program, amely írni vagy rajzolni szeretne a képernyőre, ezzel a szerverrel kommunikál.
    - Ezek a programok a kliensek.
  - A szerver mindig a lokális gépen fut.
  - Az X protokollt TCP/IP vagy DECnet fölött lehet használni.

Az X felépítése

Az X módszerének előnyei:

- a kliensnek nem kell az adott eszköz jellemzőivel foglalkoznia,
- csak a szerver kommunikációt kell implementálnunk.
- Ez pedig nem különbözik attól, mintha csak egy grafikus könyvtárat használnánk.

Az X belső működése

- **Az X protokoll (Xlib implementációja) aszinkron:**
- Az Xlib a kéréseket egy belső adatbufferben időlegesen eltárolja,
  - nem küldi egyből a szervernek (ezzel is csökkentve a hálózati forgalmat).
- A kérések a következő esetek valamelyikében kerülnek továbbításra a szerver felé:

- **1.** A kliens egy adott típusú esemény bekövetkeztére vár, de a várt típusú esemény még nem következett be,
  - nem érkezett meg a szervertől róla információ.
  - Ekkor minden kérés azonnal továbbítódik a szerver felé, hátha ezek között a kérések között van az, amelyik a várt eseményt kiváltja.

Az X belső működése

- **2.** Egyes szervertől jövő üzenetek azonnali választ igényelnek a kliens programtól.
  - Egy ilyen üzenet vételekor a kliens Xlib része azonnal válaszol a szervernek,
  - de az üzenetek helyes sorrendjének betartása érdekében az összes addigi (de még a bufferben levő) üzenetek is el lesznek küldve
- 3. Van több olyan Xlib függvény, amely rögtön üríti
  - (ilyen például az XFlush() és az XSynch()).
  - Ezeket a függvényeket akkor kell használni, ha a program hosszú ideig rajzol úgy, hogy közben nem vár semmiféle eseményt.
  - Ha ezt elfelejtenénk, akkor lehet, hogy hosszú ideig nem látszana semmi változás a szerver képernyőjén.

Az X belső működése

- **Az aszinkronitás hátrányai:**
  - nagyon nehéz az X Window rendszert használó programokat nyomon követni,
  - mert lehet, hogy valamely hibaüzenet több Xlib függvény meghívásával később jut el a klienshez.
  - Nem biztos, hogy a hibát az az Xlib függvény okozta, amely után az jelentkezett.
- A rendszer hálózat-orientált:
  - a programokban használt grafikai rutinok nem közvetlenül a képernyő memóriát manipulálják,
  - hanem a hálózaton, hálózati csomagok formájában lesznek elküldve annak a gépnek, amelynek meg kell jelenítenie az eredményt.

Az X belső működése

- **A szabvány nem tartalmaz ablakkezelő stratégiát**
- Ezek ellátása egy speciális kliens, az ablakkezelő (window-manager) feladata.
- Egy átlagos kliens működése:
  - létrehoz egy ablakot a képernyőn, azon belül létrehozhat ablakokat.
  - Az ablakkezelő megteheti azt, hogy az ablakot egy meghatározott kerettel veszi körbe, és a képernyő egész részét birtokolhatja.
- Egy X szerverhez bármennyi X kliens csatlakozhat.

Az X belső működése

- Mivel az egyes hardverek különbözőek lehetnek, a szabványnak ennek kezelésére is ki kell terjednie.
- Támogatja a mono és a színes monitorokat, 8-16-24-32 bpp színmélységig.
- A felbontásnak csak a hardvereszközök szabnak határt.
- A mutatóeszköz lehet egér, toll vagy érintőképernyő.
  - Az egerek 5 gombig támogatottak.
  - Billentyűzetből is többfélét támogat, a billentyűzetkiosztás szoftverből állítható.
- A munkaterület több monitoron is elhelyezkedhet egyszerre.

Kliensalkalmazások és ablakkezelők...

Kliensalkalmazások és ablakkezelők

- Az X-es alkalmazások készítésénél a programozók úgynevezett **widget** könyvtárakat használnak.
  - Egy widget lényegében egy előre gyártott felhasználói interfész alkotóelem
- Ezek a programkönyvtárak tartalmazzák az ablakok felületén megjelenő különböző vezérlő elemeket
  - (pl. gombok, edit mező, lista mező, stb.)
- Ezek működése eltérően lehet implementálva az egyes programkönyvtárakban,
  - ezért a programok viselkedése nagyban függ a fejlesztő által választott widget könyvtártól.

Kliensalkalmazások és ablakkezelők

- Az első *Widget* könyvtár az Athena projekt keretében kifejlesztett Athena könyvtár volt.
- Csak a legalapvetőbb elemeket tartalmazza, és a kontroll elemek kezelése eltér a manapság használatosaktól.
- Ezt követő legismertebb könyvtár az Open Software Foundation (OSF) **Motif** csomagja volt.
  - 1980-tól a korai 1990-es évekig volt igen elterjedt, de a mai napig sok helyen használják.
  - A legkomolyabb hibája, hogy súlyos összegekbe kerül a fejlesztői licence.

Kliensalkalmazások és ablakkezelők

- Manapság már vannak jobb alternatívák árban, sebességben, szolgáltatásokban.
- Ilyen például a **Gtk,** amely a **GIMP** projekthez készült.
  - Aránylag kicsi, sok szolgáltatással, bővíthető,
  - teljesen ingyenes.
- Egy másik népszerű *toolkit,* a **Qt**, amely a KDE projekt óta ismert igazán, mivel a KDE alapját szolgáltatja.
  - A Qt forráskódja nem, de a használata ingyenes.
- További alternatíva a **LessTif,** amely egy ingyenes helyettesítője a Motif-nak.
- Azonban mivel minden X kliens alkalmazás alapvetően az Xlib-en alapul, ezért egyes opciókat egységesen kezelnek.

Desktop környezet...

Desktop környezet

- Felhasználóként választhatunk ablakkezelőt kedvünk szerint.
  - a programok írói is választhatnak *widget* könyvtárakat.
- Azonban ennek a nagy szabadságnak megvannak a hátrányai:
  - A kliens program írója határozza meg, hogy milyen *widget* könyvtárat használ.
  - Azonban mivel e könyvtárak kontroll elemei nagyban különbözhetnek, előfordulhat, hogy ahány programot használunk, annyiféle módon kell kezelni.
  - Ezek a *widget* könyvtárak általában dinamikusan linkelődnek a programokhoz,
    - azonban ha a kliens programjaink különbözőket használnak, akkor az plusz memóriát igényel.

Desktop környezet

- Az ablakkezelők is sokfélék, különböző kezelési koncepciókkal.
- Az egész felület ezeknek következtében nem áll össze egy egységes egésszé,
  - például nem lehet egyben konfigurálni a kinézetét, kezelését.
- Ez azért alakult így, mert az X kialakítása során a flexibilitásra, és a felhasználók szabadságára helyezték a hangsúlyt,
  - ellentétben egyéb operációs rendszerek (például Windows, MacOS) megkötöttségeivel, zártságával.
  - Mindezért előtérbe került a komplex felület létrehozása a szabadsággal szemben.
  - Ez a helyezet vezetett az **Asztali Környezetek** (Desktop Environment) kialakulásához.

Desktop környezet

- Néhány ismertebb:

- CDE (Common Desktop Environment), KDE (K Desktop Environment), GNOME, XFCE, Enlightenment, Fluxbox, OpenBox, BlackBox, stb.

Az X indítása...

Desktop környezet

- Az X *session* elindítására két metódus közül választhatunk:
- **1. Ha a gépünk nem futtat folyamatosan X szervert**
  - egy szerver gép esetén gyakori, akkor a **startx** paranccsal indíthatjuk el és hozhatunk létre egy *session*-t.
- 2. Amikor egy **xdm *(X Display Manager)*** nevű szolgáltatás (vagy ennek egy variánsa) folyamatosan fut a gépünkön
  - Ez a munkaállomások eseten gyakori.
  - Az xdm menedzseli az X felületet.
  - Egy authentikációs szolgáltatást nyújt, ahol bejelentkezhetünk és elindítja számunka a *session-X.*

A startx működése

- A startx valójában *egy front-end* az **xinit** program számára:
  - Egy kicsit barátságosabb felületet nyújt számunkra, mint az xinit parancs formátuma.
  - Az xinit futtatja a $HOME/.xinitrc *script-et.*
  - Ha ez nem létezik, akkor a rendszerszintű xinitrc script indul el
    - (/usr/lib/x11/xinit/xinitrc vagy /etc/x11/xinit/xinitrc).
  - Ez az állomány elindíthat X kliens applikációkat és a *window manager*-t*,*
    - de jelenlegi konfigurációkban ezt az *Xclients* script teszi.
  - Az .xinitrc futatja a $HOME/.Xclients *script-et,* vagy ha ez nem létezik, akkor a rendszer szintű változatát (/etc/x11/xinit/Xclients).
  - Ennek feladata az X kliensek és a választott ***window manage**r* indítása.

Az X session indulása

- Az xdm használata esetén belépéskor hasonló folyamatok játszódnak le, mint a startx esetén.
- Az xdm létrehoz egy új session-t. Lefuttatja rendszerszintű Xsession script-et.
  - (/usr/lib/X11/xdm/Xsession vagy /etc/X11/xdm/Xsession).
- Ez a script futtatja a felhasználó $HOME/.xsession script-jét.
  - Ha nem létezik az .xsession script, akkor a $HOME/.Xclients vagy az /etc/X11/xinit/Xclients script hajtódik végre.
- Ezekben a script-ekben indíthatjuk el az X kliens applikációkat és a window manager-t

X használata távoli kliensekkel...

X használata távoli kliensekkel

- Az X architektúrájában lehetőség van rá, hogy az X szerver és a kliens program két különböző gépen fusson
  - és a hálózaton keresztül kommunikáljanak.
  - Ez az X protokoll-on keresztül zajlik.
  - Lassú kapcsolatok esetén a kommunikációt tömöríteni is lehet.
- Amennyiben használni szeretnénk ezt a funkcionalitást, akkor a következő képen kell eljárni:
  - 1. A lokális géppel (X szerver) közölnünk kell, hogy fogadja a távoli gép kapcsolódását.
  - 2. A távoli géppel (kliens program) közölnünk kell, hogy a kimenetét a lokális gépünkre irányítsa.

X használata távoli kliensekkel

- 3. Lehetséges, hogy egy gépen több X szervert használjunk,
  - vagy több felhasználó által futtatott programok egy X szerveren jelenjenek meg.
  - Ezeket is a távoli elérésnél használatos módszerekkel szabályozhatjuk.

X használata távoli kliensekkel

- Az X szerver nem fogad kapcsolódást bárhonnan.
  - Ha nem így lenne, az biztonságilag komoly problémákat vetne fel.
  - Bárki bármit odarakhatna a képernyőnkre, és monitorozhatná a billentyű leütéseinket, egér-eseményeket, stb.
- Az X kétféle authentikációs módszert támogat:
  - a host listát használó mechanizmus (**xhost**),
  - és a *magic cookie*-t használó **xauth**

X használata távoli kliensekkel

Xhost esetén:

- Ebben az esetben az X szerver egy listát kezel a kapcsolódásra jogosult gépek neveiről.
- Ehhez a listához hozzáadhatunk és elvehetünk gép neveket:

```
Xserver-host>#   xhost +xclient.valahol.hu
Xserver-host>#   xhost -xclient.valahol.hu
```

- De akár teljesen ki is kapcsolhatjuk (mindenkit engedélyezünk),
  - nem javasolt, mert az említett biztonsági problémákat veti fel.

```
Xserver-host>#  xhost  +
```

X használata távoli kliensekkel

- Visszaállítása:

Xserver-host># xhost -

- Az xhost, mint látható, nem biztonságos megoldás.
- **Oka:**
  - Egyrészt nem különbözteti meg az adott gépen lévő felhasználókat,
  - másrészt gépnév alapján dolgozik, ami félrevezethető.
  - Csak olyan hálózaton célszerű használni, ahol megbízunk a felhasználó-társainkban.

X használata távoli kliensekkel

Xauth esetén

- Az xauth azon klienseknek engedi a kapcsolódást, amelyek ismerik a titkos jelszót:
  - anthorization record vagy magic cookie.
- A *cookie*-k az ~/.Xauthority állományban tárolódnak.
  - Vigyázni kell a jogok beállításánál, hogy csak a tulajdonos olvashassa.
  - Ezt az állományt az xauth program kezeli.
- Az xauth protokoll használatához az X szervert az -auth <authfile> paraméterrel kell indítani.
  - Az authentikációs kommunikáció továbbra is kódolatlanul zajlik a hálózaton,
  - ezért a biztonság szempontjából érdemes lehet egy kódolt csatornával kiegészíteni: ssh

Köszönöm a figyelmet!