

SuperKaramba újratöltve

A SuperKaramba már többször szerepelt a Linuxvilágban, ezért most inkább azokhoz szólnék, akik az érdekességet nem a projekt felélesztésében vélik felfedezni, sokkal inkább az egyedi, látványos és hasznos témák kreálásában.

Mint a bevezetőben leszögeztem, ebben a cikkben nem célom a közkedvelt „Karamba” bemutatása. Még a felélesztésre sem térek ki: a sokszor bemutatott és csontig rágott témával senkit nem szeretnék feleslegesen untatni. A szokatlan bevezetést egy huszáros vágással le is zárom: a továbbiakban feltételezem, hogy az Olvasó (kedvenc disztribúciójának csomagkezelőjével, esetleg forráskódból) már feltelepítette ezt a kiváló programot, majd felhasználóként kiadta a `superkaramba` parancsot. A művelet eredményeképpen, néhány pillanat múlva a szoftver már kérte is a megnyitandó `*.theme` leíró állományokat.

Szép, szebb, leghaszontalanabb

Amikor néhány éve először kutakodtam *SuperKaramba* témák után, némi iróniával „rosszmájú” következtetéseket vontam le: voltak szép munkák, akadtak szebbek, és teljesen felesleges szenzorokkal felszerelt giccses darabok. A méltán híres <http://kde-look.org> honlap ide vonatkozó választékából az egyik kezemen meg tudtam volna számolni az értelmes munkákat. Nem tudtam mire vélni a jelenséget, csak csóváltam a fejem: „Létezik olyan ember, aki azt szereti, ha az aktuális időjárást a monitorról olvassa le ahelyett, hogy kinézne az ablakon? Ennyire gyakori, ha valakinek kevés a rendszertálcán lévő óra, ezért egy éktelen nagy időmérőt tesz a képernyőre második eszközként?” De a legjobb csak ezután jött! Egy rendszerparaméter bűbájos kijelzésétől majdnem „üvöltve futottam ki a világból”: az alkotó neon-sárga

színű információkat tett ríkító piros háttérre. Utóbbi attrakciót hosszú ideig egy nagyon rossz tréfának véltem, miszerint talán olyan szándékkal született, hogy a felhasználó szeme órákon át káprázzon tőle.

Végül aztán rádöbbentem, nincs ezekben semmi különös, csupán nem vagyunk egyformák. Triviális felismerés volt, de ahogyan két ízlés egyezése annyira valószínű, mint két azonos pofoné, úgy más irányban kerestem a megoldást. A szubjektív megítélés szülte elégedetlenségemet munkába fojtottam, és csináltam magamnak egy olyan „Karamba-témát”, amitől nem rázott ki a hideg. Ez a hajlandóság szerencsére mind a mai napig meg van bennem, mára azonban kissé más megközelítésben: az előbb leírt linket aktívan használom, bizonyos kreatörök *GPL* munkáit keresve. Ezeket a műveket aztán jellemzően átalakítom – saját ízlésemnek megfelelően.

Mire kell törekedned?

Nehéz kérdés! A választ nem is tudnám úgy megadni, hogy ne szóljon közbe a részrehajlás ténye. Inkább olyan megoldást választok, ahol leírom, hogy én milyen szempontok szerint képzelem a stílusos és hatékony *SuperKaramba* környezetet. Reményeim szerint ebből az Olvasó le tudja majd szűrni a tanulságokat, aminek hatására (nézeteim helyeslése vagy ellenzése által) szilárdulhat a véleménye – feltéve, hogy még csak „tapogatózik”. A hatékony munkaasztal megítélése a felhasználó beállítottóságától függ. Van, aki esküszik a villámgyors *BlackBox*-ra, van aki a *GNOME*-ban

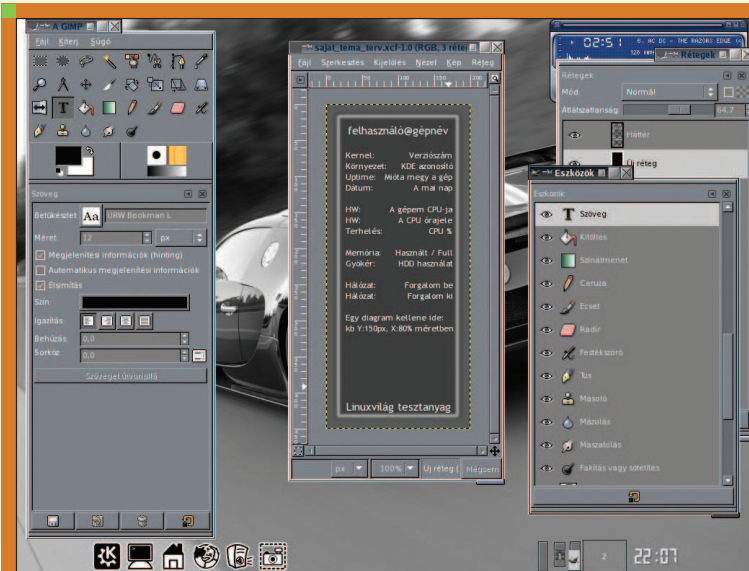


1. ábra A munkaasztalom, „pucéron”

talál rá a számításaira, hogy a többi felületet ne is említsem. Mindenki másképp csinálja: ismerősen cseng az öregecske hazai sláger – és kicsoda igazságot rejt! Én sem vagyok kivétel: évek óta *KDE*-párti vagyok, az „elefánt méretű” környezet a kezdetektől fogva lenyűgöz engem is. No nem a méretével – nem is az igényeivel, sokkal inkább az összetett képességeivel: az extráktól roskadó asztal mérhetetlenül sok segédprogramot és lehetőséget biztosít a rálátásomhoz mérten. Így a példaként bemutatandó téma felépítésénél a *KDE* asztalom megjelenését tartottam szem előtt...

Íme, a formák és a színvilág

Személy szerint leginkább a fehér, a fekete és a kék színekkel vagyok kibékülve – ezeket (és árnyalataikat) huzamosabb ideig is el tudom viselni a képernyőn. Pontosítanék: kizárólag csak ezeket tudom elviselni... A mindenható eléggé kényes látást „programozott” nekem: sajnos még egy csodaszép, színes természetfotó is zavaróan hat rám hosszú távon, így a tetszetős munkákat legtöbbször telítetlené teszem.



2. ábra Készül a téma, épül a skicc

Az asztalom „alapját” rendszerint a következőképpen alakítom ki: három-négy szürkeárnyalatos képet bemásolok egy mindenki által olvasható (e célra fenntartott) mappába, majd készítek egy apró szkriptet, ami a grafikákat meghatározott szisztéma szerint „körkörösén” átnevezi a számítógép indulásakor. A KDE asztal hátterét aztán az egyik ilyen grafika tallózásával határozom meg, így minden egyes indítás után más háttérképpel fogad a profilom. Igyekszem az ikonsettekben is követni a kissé különözlésvilágot, tehát a két színre szűrt remekműveket favorizálom. Természetesen a menüszerkezetet és a konzolokat is ehhez igazítom, miközben bizonyos fókú áttűnésre törekszem.

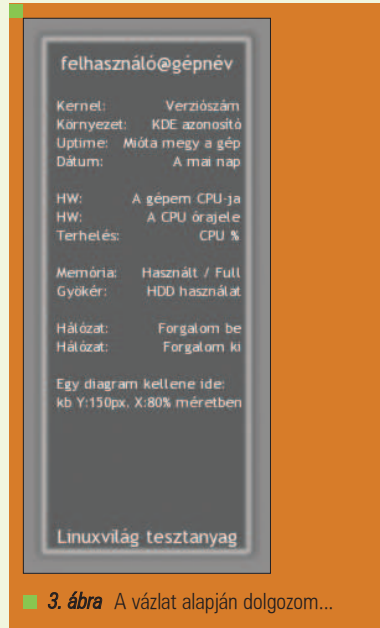
Külsőin...

Nos, egy ilyen „szürke világra” nem nehéz egyszerű és elegáns *SuperKaramba* témát készíteni, hiszen csak a színekben csökkentett környezet szabályait kell figyelembe venni. Képzelnék el egy képet, ami az egész vizualitás alapját fogja nyújtani, lehetőleg áttetsző háttérrel. Nyúlunk bátran a *GIMP*-ért! Kezdjük nagyjából $200 \times 450 \text{px}$ méretű áttetsző rajzfelületen, ennek sarkait kerekítsünk le, és lágy szélekkel öntsük ki a területet fekete színnel. A sötétre festett réteg áttetszőségét állítsuk hatvan százalékosra, majd egy új rétegen készítsünk egy fehér színű keretet.

Az eredményt ezután (összefésülve) el kell mentenünk tetszőleges névvel, mondjuk *PNG* formátumban (esetemben *sajat.png*). Ezzel a résszel meg is volnánk, de még ne zárjuk be a képmanipulátort: a műveletek közben szerepe lesz!

A belbecs ereje

A történet most kezdődik igazán: a stílusos munka belbecsben is meg kell nyilvánuljon. Márpedig a lekérdezett (és kijelzett) paraméterek terén szükséges némi rálátás a *Linux/Unix* parancsokra, a *SuperKaramba* dokumentációjában leírt megoldások mellett. Utóbbi leírás egyébként nem szűkölködik a lehetőségek ismertetésében! Hálózati és rendszer információk, merevlemez „monitorozás”, egyéb programvezérlések egyaránt beleszőhetők a témába. Az elképzelt megoldásban a hálózat, a merevlemez és a memória adatait fogjuk előtérbe helyezni, emellett a rendszer paraméterei (kernel, felhasználó, gépnév, asztali környezet) is szerepet kapnak majd. A programvezérléshez nem fogunk görcsösen ragaszkodni, de az *XMMS* alapfunkcióit például integrálhatnánk... No, mindegy, erre a részre azért még visszatérek! Az előző blokkban leírtam, hogyan lehet egyszerűen elkészíteni a munka grafikai alapját. Tényleg nem volt nagy dolog, viszont most komolyan végig kellene gondolni az alkotás megjelené-



3. ábra A vázlat alapján dolgozom...

sét: a *GIMP* még fut, így egy újabb rétegen érdemes lehet kipróbálni néhány lehetőséget, szövegblokkokat létrehozva és türelmesen rendezgetve (valójában a kész témát szeretnénk előzetesen modellezni). Az elfogadhatónak ítélt skiccet hagyjuk nyitva! A vázlaton szereplő, megjelenítendő paramétereket rögzíteni kell a *SuperKaramba* új témájának fő állományában. Tetszőleges szerkesztővel hozzunk létre egy szöveges fájlt, mondjuk *sajat.theme* néven! Első sorát (saját példám szerint) kezdjük így:

```
karamba x=30 y=30 w=200 h=450
```

Ez a következőket jelenti: „*Karamba*” témáról van szó, aminek alapértelmezett elhelyezkedése az asztal „*x,y*” pozíciójába kéretik, a kijelzőn lévő információk helyzetét pedig egy „*w,h*” pixelméretű mátrixon fogjuk megadni (ez rendszerint egyezik a létrehozott **.png* háttérkép dimenzióival). A következő két sor így fest:

```
defaultfont color=255,255,255
↳ fontsize=9 font="Tahoma"
↳ shadow=2 bgcolor=0,0,0
image x=0 y=0 path="sajat.png"
↳ name="background"
```

Ezzel azt közlöm, hogy az alapértelmezett szöveg színe fehér lesz (*RGB 255,255,255*), típusa *Tahoma*, kilences méretben. A betűknek vastag fekete



4. ábra Az asztal bal oldalán a kész téma!

árnyékot szeretnék (RGB 0,0,0)! A téma grafikai alapját az előzőekben készített egyszerű *sajat.png* kép fogja biztosítani, amit a pixelmátrix $x,y(0,0)$ pontjába kell elhelyezni (így a mátrix és a PNG grafika pontosan „lefedí” egymást). A bevezetés ezzel meg is volna, következzenek kijelző alkotóelemei.

A GIMP-pel létrehozott skicc első szöveges sorában a büszke felhasználó és a számítógép neve „feszít”. Nagyon fontos, hogy ez a szöveg milyen pozícióban helyezkedik el! Váltunk át a képmanipulátorra, és vigyük a mutatót a rétegelt kép első gépelt sora elé. A szerkesztő ablak bal alsó sarkában leolvasható a pontos pozíció: ebben ez $x,y(42,40)$ érték. Rögzítsük ezt is a *theme* fájlban, tehát a következő sornak nagyjából így kell kinéznie:

```
text x=42 y=40 sensor=program
↳ program="echo ${USER}@
↳ `hostname`" fontsize=14
```

Ugyebár említettem, némi gyakorlat szükséges a Linux parancsok terén: az echo parancs az aktuális szintaktikában itt a *felhasználó@gépnév* kimenetet adja eredményül. A pozicionálás nem szorul magyarázatra, viszont a *sensor* szó igen: így kell jelölni azokat az értékeket, melyek nem általunk rögzített szövegek, hanem „szenzorszerűen” kiolvasott adatok. A *program* nevű vezérlő pedig arra utal, hogy egy külső programból nyerjük ki az adatokat,

nem pedig a „Karamba” beépített lehetőségeit használjuk (e külső program jelen esetben az említett *echo*). A használandó betűtípus méreténél csúnyán eltértem az alapértelmezettől – azért, hogy ez a sor azonnal szemet szűrjön mindenkinek!

Ettől kezdve kicsit kuszább lesz a helyzet, mivel a kijelző ugyanazon sorában lesz a szenzoros érték és az általunk megadott szöveg is, a példában a „Kernel:” felirat, rögtön utána a mag jelölése. Ezt a következőképpen tudjuk megoldani:

```
text x=30 y=76 value="kernel:"
text x=116 y=76 sensor=program
↳ program="uname -r"
```

A pozicionálás itt sem érdemel különösebb értelmezést, hiszen a GIMP folyamatosan „sűg”: a magassági értékek értelemszerűen egyeznek, „x” irányban azonban el vannak tolvá. Az első sorban olvasható *value* jelölés szolgál arra, hogy a „Kernel:” felirat statikus legyen. Eközben az *uname -r* parancs kiolvassa a mag verziószámát. Most több elképzelést valósítsunk meg „egy körben”!

```
text x=30 y=91
value="környezet:"
text x=140 y=91 sensor=program
↳ program="kde-config -version
↳ | grep KDE | sed -e 's/.*:
↳ //'"
```

```
text x=30 y=106 value="Uptime:"
text x=102 y=106 sensor=uptime
↳ format="%Hh:%Mm:%Ss"
↳ interval=1000
text x=30 y=121 value="Dátum:"
text x=104 y=121 sensor=time
↳ format="yy.MM.dd ddd"
↳ interval=60000
```

Nos, így kell lekérdeznünk a KDE környezet verzióját, majd a *sed* által érdekes levágunk az információ felesleges részét. Együttal megmérjük a bekapcsolás óta eltelt időt, hogy adott formátumban ki tudjuk jelezni (ehhez nem kell külső program, így itt kihagytam a programvezérlőt). Az *uptime* szenzor frissítését 1000 ms-os (tehát egy másodperces) időközökre állítjuk, az *interval=1000* érték segítségével. Végül, de nem utolsó sorban a dátum is a témára fog kerülni, amit percenként leellenőrzünk (ezen a ponton kérhetnénk jóval lomhább frissítést is, viszont egy esetleges éjféle „átfordulás” esetén a kijelző csúnyán késne). A következő blokk a processzorra, a gyökér partícióra és a memória foglaltságára fog vonatkozni. Kezdjük a központi egységgel! A CPU adatainak lekérdezéséért egyaránt felel a „külsős” *cpuinfo* és az *uname -p* parancs, valamint a *SuperKaramba* beépített CPU szenzora. A terhelés mérése másodperces ciklusokban esedékes, az órajel kijelzése fél percenként frissül:

```
text x=30 y=150 value="Hw:"
text x=126 y=150 sensor=program
↳ program="uname -p"
text x=30 y=165 value="Órajel:"
text x=120 y=165 sensor=program
↳ program="cat /proc/cpuinfo |
↳ grep 'cpu MHz' | sed -e
↳ 's/.*: //' | sed -e
↳ 's/\.../MHz/'"
↳ interval=30000
text x=30 y=180
↳ value="Terhelés:"
text x=148 y=180 sensor=cpu
↳ format="%v%" interval=1000
```

Ezek után már semmiség a gyökér és a központi tár mérése, kijelzése. Erre a célra szintén a beépített szenzorokat használhatjuk legegyszerűbben:

```
text x=30 y=212
↳ value="Memória:"
```

```
text x=98 y=212 sensor=memory
↳ format="%um / %tm Mb"
text x=30 y=227 value="Gyökér"
↳ foglalt:"
text x=146 y=227 sensor=disk
↳ format="%up%" mountpoint="/"
```

A hálózati vezérlő sem maradhat ki a listából, a befelé és a kifelé mutató adatforgalom sebességét szeretném monitorozni:

```
text x=30 y=260 value="Hálózat"
↳ (1e):"
text x=128 y=260 sensor=network
↳ device="eth0" format="%in
↳ kB/s" interval=2000
↳ decimals=1
text x=30 y=275 value="Hálózat"
↳ (fe1):"
text x=128 y=275 sensor=network
↳ device="eth0" format="%out
↳ kB/s" interval=1000
↳ decimals=1
```

Most már csak egy dolog van hátra. Egy szép, aktív grafikont gondoltam ki a példa elején, ami a *CPU* terhelését mutatja folyamatában (a szándékot a skiccen megjelenítettem). Erre is van beépített lehetőség, amit a következők szerint kell használni:

```
graph x=50 y=312 w=100 h=80
↳ sensor=cpu points=100
↳ color=255, 255, 255
```

Tehát az *x,y* koordinátákra kérünk egy fehér színű aktív grafikont, ami *100x80* pixel szélességben *100* lehetséges értéket tud megjeleníteni. Persze az analízátor jobban mutat, ha van mögötte egy erre a célra készült mutatós háttérgrafika. Ezt most nem részletezem, a *GIMP* segítségével gyorsan összeütök egy *130x100* pixel méretű gráfot. Ha elkészült, akkor a *sajat.theme* három bevezető sora után beszúrom negyediknek a friss-ropogós képet:

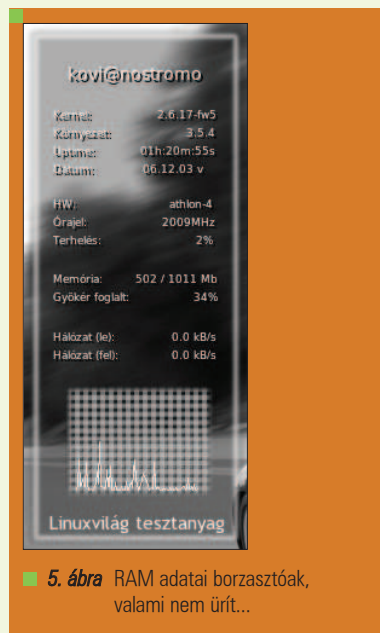
```
image x=36 y=300
↳ path="racs.png"
```

Nos, ha mindennel végeztünk, akkor a létrehozott két képet, valamint a *sajat.theme* állományt mozgassuk ugyanabba a mappába – majd a *superkaramba* parancs kiadása után tallózzuk ki a munkánkat. Remélem, tetszik a letisztult a téma. A pozicioná-

lás pontosítása ugyan még hátravan, de ez már teljesen személy- és asztal függő. Így ha valakinek nem tetszik egy kijelzett adat helye, akkor az *x,y* koordinátákon változtasson bátran. A változtatások idejére azonban az aktuális témát mindig be kell zárni, különben a *SuperKaramba* (bizonyos verziókban) futása hibával megszakad. Említettem, hogy akár az *XMMS* vezérlését is rátehetnénk az elkészült műre. Engedelmetekkel nem követném el ezt a „bűnt” (nem látom értelmét), de aki mindenképpen szeretné megoldani, annak először is készítenie kell egy gombsor grafikát. Olyan képet kell szerkeszteni, ahol a népszerű lejátszó alapfunkcióihoz (előző szám, lejátszás, állj, következő szám) tartozó gombok egymás mellett sorakoznak, a téma szélességéhez igazodva. Nevezük a képet *xmms_irany.png* néven! Ekkor a vezérlést biztosító sorok valahogy így festenek:

```
image x=60 y=400
↳ path="xmms_irany.png"
clickarea x=60 y=400 w=20 h=20
↳ onclick="xmms -rew"
clickarea x=90 y=400 w=20 h=20
↳ onclick="xmms -play"
clickarea x=120 y=400 w=20 h=20
↳ onclick="xmms -stop"
clickarea x=150 y=400 w=20 h=20
↳ onclick="xmms -fwd"
```

A sorok működése rém egyszerű: meghívjuk a gombsoros képet *x,y(30,400)* pozícióba, majd a *WEB* kapcsán ismerős *Imagemap* eljáráshoz hasonlóan területeket definiálunk. Az első aktív sor így értelmezhető: a *x,y(60,400)* ponttól jobbra, lefelé irányban *20x20px* méretű területen figyeljük az egérgomb lenyomását. Ha megtörténik, akkor kiadjuk az *xmms --opcio* parancsot (mivel a lejátszó futása például egy konzolról is változtatható a leírt utasításokkal). A teljes igazság azonban tartozik még két apró információ, amivel tartozom. Az egyik dolog egy fontos képesség „elhallgatása”: amennyiben a témát tartalmazó mappában létrehozunk egy (bizonyos szabályok szerint felépülő) *Python* szkriptet, akkor az interaktivitás magasabb szintre emelhető. Például a kész kijelzőn jobb egérgékket nyomva adott dolog történhet: akár a háttérgrafika cseréje,



5. ábra RAM adatai borzasztóak, valami nem ürt...

vagy éppen a teljes kinézet átváltozása. Amennyiben igény mutatkozik rá, akkor vissza fogunk majd térni erre egy későbbi számban – ahol egy nagyon összetett, sok blokkos, több ablakos interaktív téma létrehozásával fogok megpróbálkozni.

A másik apróság arról szól, hogy a létrehozott munka természetesen egyszerű példa jellegű. Személy szerint a cikk elején olvasható linkről töltem le a nekem tetsző remekműveket, majd saját ízlésem szerint „tombolom szét” őket, hogy a fekete-fehérre szűrt asztalomhoz illeszkedjenek...

Aki kedvet érezne saját munkák alkotáshoz, miközben nem szeretné begépelni a leírt sorokat (vagy éppen nem áll kezére a *GIMP*), látogasson el fél percre a <http://kovi.uw.hu/lo2007> címre! Ide töltöttem fel a cikkben szereplő grafikákat nyers formákkal, és az aktuális *theme* állományt – segítségképpen. Egyet kérek szépen mindössze (mondjuk úgy, cserébe): ha valaki alkotott egy tetszetős és hasznos *SuperKaramba* témát, azt ne felejtse el önzetlenül megosztani a többiekkel, tehát irány a <http://kde-look.org/>! Tartalmas kikapcsolódást!

Kovács Zsolt (kovi@linuxforum.hu)

Quake fanatikusan. Négy éve a debreceni linuxosok egyike. Töretlenül hisz a Slackware terjesztésben.