



Gömülü GUI Teknolojileri

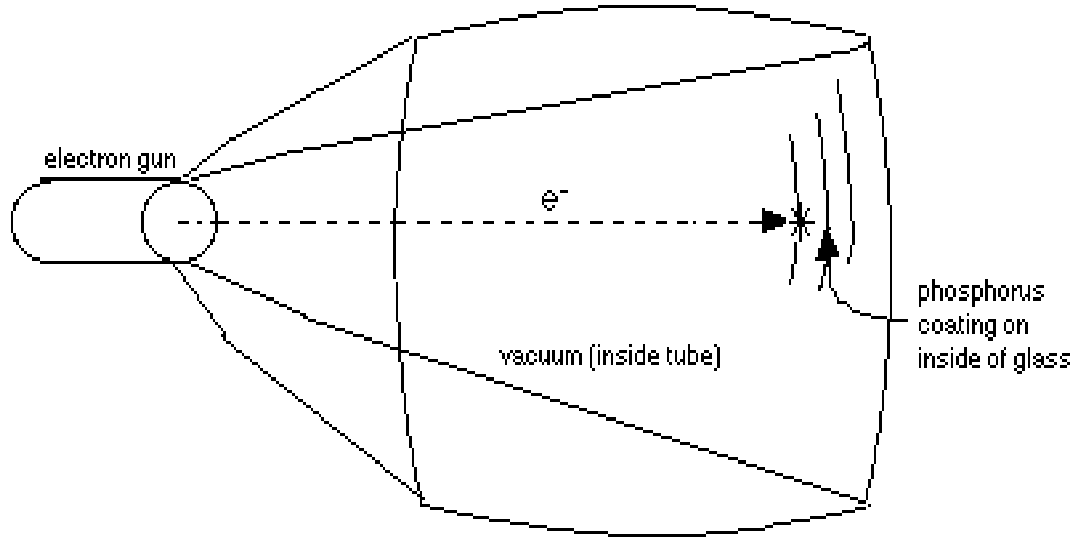
Derya Sezen

Eliar A.Ş.

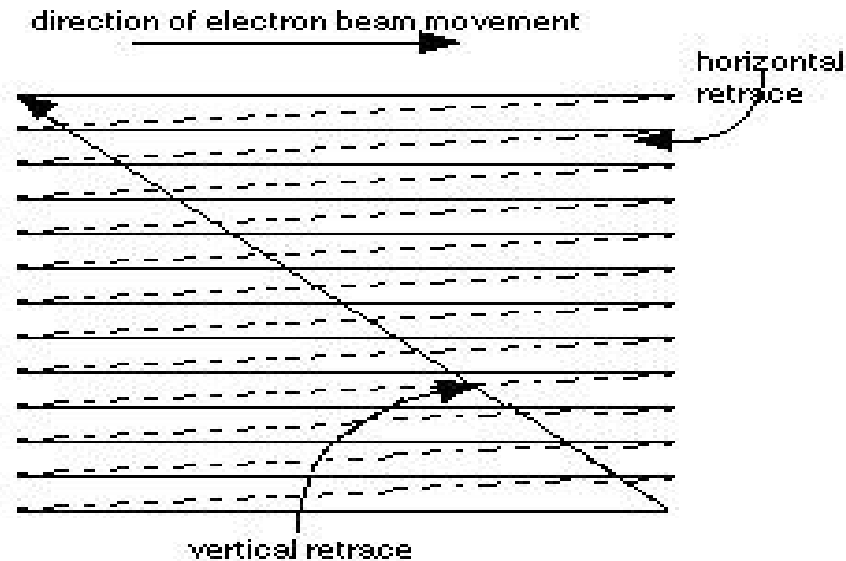
derya.sezen@eliar.com.tr

- Video Modu Zamanlamaları (Video Mode Timings)
- MTRR (Memory Type Range Register)
- Linux FrameBuffer Device
- FrameBuffer Programlama
- Gömülü GUI Teknolojileri

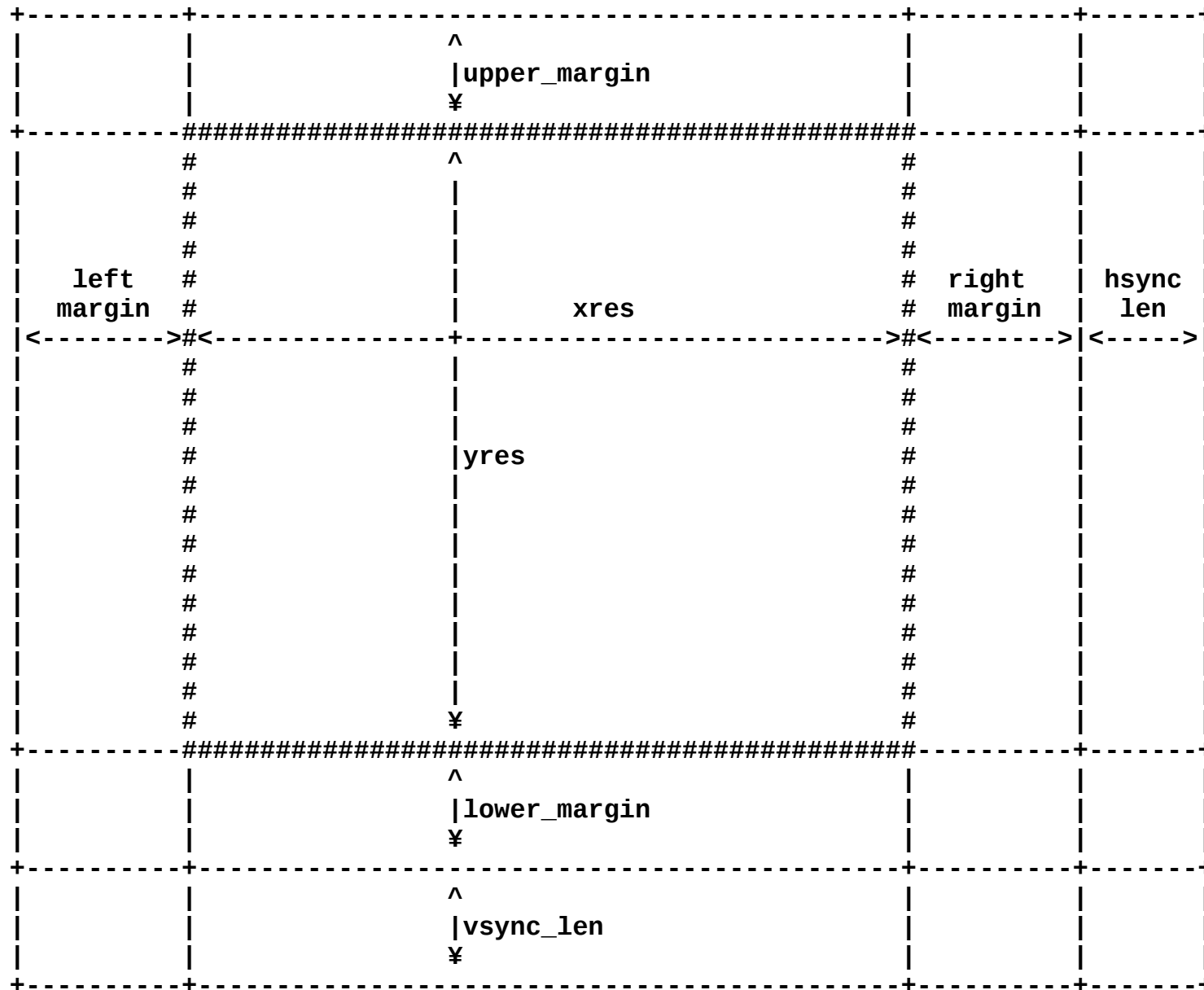
- Video Modu Zamanlamaları (CRT) :



- Elektron ışınları elektron tabancasından çıkarak, ekranın üzerinde bulunan fosfora çarpar, fosfor foton yayar ve görünür hale gelir.
- Elektronların yoğunluğu azaltılarak veya arttırılarak ekran parlaklığı ayarlanabilir



- **Horizontal Retrace** : Her satır bittiğinde elektron tabancasının bir sonraki satırın başına gitmesi işlemidir.
- **Vertical Retrace** : Ekranın sol alt köşesine gelindiğinde, ekranı tekrar taramak için tabancanın sağ üst köşeye gitmesi işlemidir.



- Sabit frekanslı monitorler (VGA)

$hfreq = 31,5 \text{ kHz}$.

640x480 için $dotclock = horiz_total * hfreq$
 $dotclock = 20,16 \text{ Mhz}$.

$horiz_total = width + right_border + sync + left_border$
 $\approx width * 1,2$
-> $dotclock \approx 24 \text{ Mhz}$.

Kartın desteklediği frekans $25,175 \text{ Mhz}$.
-> $horiz_total = 25,175 / 31,5$
 ≈ 800

$\text{horizontal_retrace} = \text{right_border} + \text{sync} + \text{left_border} = 160$

$\text{sync} \approx \frac{1}{2} \text{horizontal_retrace}$

$\text{right_border} \approx \frac{1}{2} \text{left_border}$

$\rightarrow \text{sync} \approx 80, \text{right_border} \approx 24, \text{left_border} \approx 56$

“640x480”	25,175	640	664	744	800	???	???	???	???
	DCF		HR	SH1	SH2	HFL			

$\text{left_margin} = \text{HFL} - \text{SH2}$

$\text{right_margin} = \text{SH1} - \text{HR}$

$\text{hsync_len} = \text{SH2} - \text{SH1}$

Vertical Timing :

vfreq 480 satır için VGA 'da 60 hz.

$hfreq = vfreq * vert_total$

$\rightarrow vert_total = 525 \approx height * 1.1$

“640x480”	25,175	640	664	744	800	480	495	497	525
	DCF	HR	SH1	SH2	HFL		VR	SV1	SV2 VFL

$upper_margin = VFL - SV2$

$lower_margin = SV1 - VR$

$vsync_length = SV2 - SV1$

- Çoklu Frekanslı Monitörler (Multiscan Monitors)

hfreq : 31-95 kHz.

vfreq : 50-130 Hz.

640*480 (pratikte 768*528) için maximum tazeleme oranı :

(vtotal \approx width * 1.1 , htotal \approx height * 1.2)

$$\begin{aligned} \text{hfreq} &= \text{vfreq} * \text{vtotal} \\ &= 68.6 \text{ kHz.} \end{aligned}$$

$$\begin{aligned} \text{pixclock} &= \text{hfreq} * \text{htotal} \\ &\approx 52.7 \text{ Mhz.} \end{aligned}$$

-> Grafik arabirim 50 Mhz. , 60 Mhz. de çalışabiliyor

1. 50 Mhz. seçilir : Refresh rate biraz düşer ($\sim 5\%$)

2. 60 Mhz. seçilir : Monitör özelliklerini aşabilir, border ekleyerek sorun çözülebilir.

- MTRR (Memory Type Range Register)

- Intel P6 ailesi (Pentium Pro, Pentium II ve sonrası) ile kullanılmaya başlandı
- Cyrix 6x86, 6x86MX ve MII işlemcilerinde MTRR'a benzer yapıda olan ARR (Adress Range Register) bulunur.
- AMD Athlon ailesi işlemcilerde Intel tarzı MTRR kullanılmaya başlanmıştır.

- `cat /proc/cpuinfo | grep flags`

flags : fpu vme de pse tsc msr pae mce cx8 apic sep **mtrr** pge mca cmov
pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm

```
Processor type and features
Menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Press
features. Press <Esc><Esc> to exit, <?> for Help. Legend: [*] built-in [

[*] Low latency scheduling
(Pentium-Pro/Celeron/Pentium-II) Processor family
[*] Machine Check Exception
CPU Frequency scaling --->
<M> Toshiba Laptop support
<M> Dell laptop support
<M> /dev/cpu/microcode - Intel IA32 CPU microcode support
<M> /dev/cpu/*/msr - Model-specific register support
<M> /dev/cpu/*/cpuid - CPU information support
[ ] E820 proc support
<M> BIOS Enhanced Disk Drive calls determine boot disk (EXPERIMENTAL)
(4GB) High Memory Support
[*] HIGHMEM I/O support
[ ] Math emulation
[*] MTRR (Memory Type Range Register) support
[ ] Symmetric multi-processing support
[ ] Local APIC support on uniprocessors
[ ] Unsynced TSC support
```

Kernel özelliklerinde CONFIG_MTRR 'ı açarsak /proc/mtrr 'ın oluşmasını sağlarız.

- PCI veya AGP veriyolu üzerindeki bir ekran kartı ile kullanımı etkili
- Modları : UC(Uncacheable), WR(Write Combining), WT(Write Through), WB(Write Back), WP(Write Protect)
- “write-combining” özelliği açılarak PCI veya AGP veriyoluna olan “write” işlemleri birleştirilerek kombine bir şekilde gerçekleştirilir.
- WC, 32 Byte'lık Buffer Kullanır
- 2,5 kat veya daha hızlı imaj yazma hızı

- /proc/mtrr : 2 adet arabirimi bulunur :

1. Okuma ve yazma yapılabilen ASCII arabirimi

2. ioctl() arabirimi

- MTRR'lara bakmak için :

```
cat /proc/mtrr
```

```
reg00: base=0x00000000 ( 0MB), size= 512MB: write-back, count=1
```

```
reg01: base=0xd8000000 (3456MB), size= 64MB: write-combining, count=1
```

- MTRR yaratmak için :

```
# echo "base=0xf8000000 size=0x400000 type=write-combining" >| /proc/mtrr
```

```
# cat /proc/mtrr
```

```
reg00: base=0x00000000 ( 0MB), size= 512MB: write-back, count=1
```

```
reg01: base=0xd8000000 (3456MB), size= 64MB: write-combining, count=1
```

```
reg02: base=0xf8000000 (3968MB), size= 4MB: write-combining, count=1
```

- Eklediğimiz satır 0xf8000000 adresinde bulunan 4 Mbyte.'lık vadeo RAM içindi. Bu adresi bulabilmek için X Server'ın loglarında aşağıdakine benzer bir satır bulacaksınız :

(--) NV(0): Linear framebuffer at 0xD8000000

Gene X log'larından framebuffer 'un büyüklüğünü öğrenmek için :

(--) NV(0): videoram: 65536k

- Overlap MTRR yaratmak :

```
# echo "base=0xfb000000 size=0x1000000 type=write-combining" >/proc/mtrr
```

```
# echo "base=0xfb000000 size=0x1000 type=uncachable" >/proc/mtrr
```

```
# cat /proc/mtrr
```

```
reg00: base=0x00000000 ( 0MB), size= 64MB: write-back, count=1
```

```
reg01: base=0xfb000000 (4016MB), size= 16MB: write-combining, count=1
```

```
reg02: base=0xfb000000 (4016MB), size= 4kB: uncachable, count=1
```

Bazı Voodoo grafik kartları alanın başındaki bu 4 kb. lık alanı register'lar için kullanıldığı için ihtiyaç duyuyor.

- MTRR silmek :

```
# echo "disable=2" >| /proc/mtrr
```

• Linux Frame Buffer Device

- Grafik arabirim için bir soyutlama mekanizmasıdır.
- Yazılımların grafik arabirime, alt taraftaki donanım register gibi detaylarla uğraşmadan, direk erişebilmesini sağlar.
- Her platform için generic bir arabirim imkanı

/dev/fb0 # 1. framebuffer aygıtı

/dev/fb1 # 2. framebuffer aygıtı

...

/dev/fb31 # 32. framebuffer aygıtı

Eskiye uyumluluk için :

/dev/fb0current -> fb0

/dev/fb1current -> fb1

- Örnek olarak X Server öntanımlı olarak /dev/fb0 'ı kullanır.

\$FRAMEBUFFER 'a farklı bir değer atanarak bu değiştirebiliriz :

export FRAMEBUFFER=/dev/fb1

- Ekranın bir snapshot'unu almak için :

cp /dev/fb0 snapshot

dd if=/dev/fb0 of=snapshot obs=WIDTHxHEIGHTxBPP

- Çözünürlük :

- “fbset” : Ekranın video modu işlemleri

```
#fbset -i  
mode "1024x768-76"  
# D: 78.653 MHz, H: 59.949 kHz, V: 75.694 Hz  
geometry 1024 768 1024 768 16  
timings 12714 128 32 16 4 128 4  
rgba 5/11,6/5,5/0,0/0  
endmode
```

Frame buffer device information:

```
Name      : VESA VGA  
Address    : 0xd0000000  
Size       : 67108864  
Type       : PACKED PIXELS  
Visual     : TRUECOLOR  
XPanStep   : 0  
YPanStep   : 0  
YWrapStep  : 0  
LineLength : 2048  
Accelerator : No
```

- /etc/rc.* veya /etc/init.d/* ye tanımlanabilir

• FrameBuffer Programlama

- /dev/mem ile aynı konseptte, normal bir memory gibi, tek farkı video kartının memory'si olması ve user-space'den direk erişilebilmesi için mmap edilmesi.
- read(), write(), seek(), mmap()
- /dev/fb* üzerinde ioctl() :
 - donanım ile ilgili data query işlemleri
 - data set işlemleri

- <linux/fb.h> dosyasında bulunmaktadır. Başlıca :

- Donanım ile ilgili read-only bilgilerin sorgulanması: isim, ekran hafıza organizasyonu, adresi, uzunluğu...

- Değiştirilebilir özellikler : Derinlik, Geometrisi, zamanlama...
Bu gibi değerleri değiştirmeye çalıştığınızda sürücü bazı değerleri ekran kartı yapısı gereği yuvarlayabilir. Eğer değiştiremiyorsa EINVAL döndürür.

- Renk haritası üzerindeki işlemler : getpixel() , setpixel()

- Coding :

```
// framebuffer dosyasını yazmak ve okumak için açıyoruz
```

```
fbfd = open("/dev/fb0", O_RDWR);
```

```
// fb.h daki tanımlarla ioctl ile donanım bilgilerini (fixed info) alıyoruz
```

```
ioctl(fbfd, FBIOGET_FSCREENINFO, &finfo);
```

```
// Değişken ekran bilgilerini alıyoruz (variable info )
```

```
ioctl(fbfd, FBIOGET_VSCREENINFO, &vinfo)
```

```
// Ekran Boyutunu Byte cinsinden hesaplıyoruz
```

```
ekran_boyutu = vinfo.xres * vinfo.yres * vinfo.bits_per_pixel / 8;
```

```
char *fbp = 0;  
// framebuffer device'ı memory'ye map ediyoruz ( lineer alan )  
fbp = (char *)mmap(0, ekran_boyutu, PROT_READ | PROT_WRITE,  
MAP_SHARED, fbfd, 0);  
  
// x,y koordinatlarının yerini bulmak için  
konum = (y+vinfo.yoffset) * finfo.line_length +  
        (x+vinfo.xoffset) * (vinfo.bits_per_pixel/8);
```

// Hangi BPP 'de işlem yapacağız ?

// 32 BPP ise RGBT (Red-Blue-Green-Transparency)

```
if ( vinfo.bits_per_pixel == 32 ) {
```

```
    *(fbp + konum) = 100 ; // Blue
```

```
    *(fbp + konum + 1) = 15; // Green
```

```
    *(fbp + konum + 2) = 200; // Red
```

```
    *(fbp + konum + 3) = 0; // Transparency yok
```

```
}
```



```
// 16 BPP ise RGB 'yi 16 bite bölüştürmemiz gerek
```

```
// | 5 bit Red | 6 bit Green | 5 bit Blue |
```

```
if ( vinfo.bits_per_pixel == 16 ) {
```

```
    int b = 10;    // Blue
```

```
    int g = 20;    // Green
```

```
    int r = 25;    // Red
```

```
    unsigned short int t = r<<11 | g << 5 | b;
```

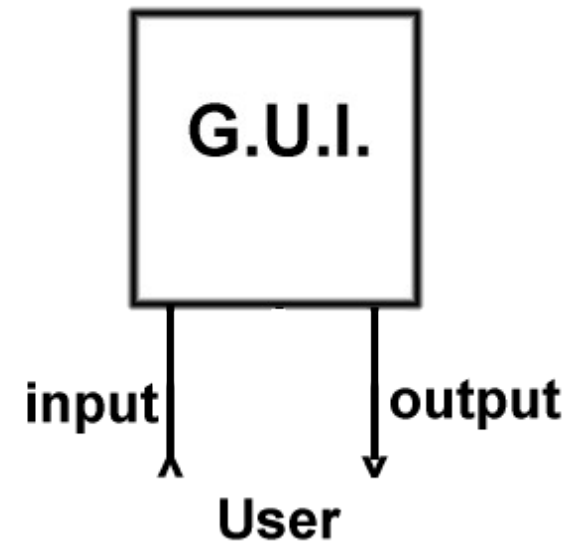
```
    *((unsigned short int*)(fbp + konum)) = t;
```

```
}
```

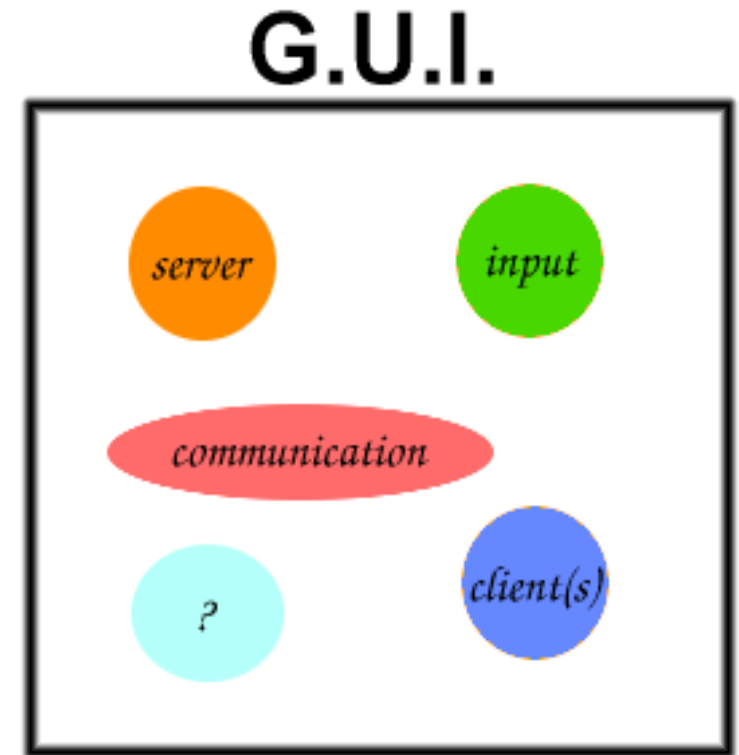
```
munmap(fbp, ekran_boyutu);
```



- GUI 'ye giden yoldaki diğer taşlar :
- Input :
 - keyboard, touch devices...
 - Kernel 2.6 Event Sistemi , keywatcher
- Output :
 - Crt, lcd, plasma, dotmatrix...



- Communication
 - Tcp, uds (unix domain sockets)
 - Shm (shared memory)
 - ...
- Server
 - Event'lerin yakalanması, ilgili client'e verilmesi
 - Client'lara koordinat bilgilerinin verilmesi
 - ...
- Client(s) :
 - Ekran değişikliklerin yansıtılması
- Senkronizasyon
 - Mutex, semaphores



- Gömülü GUI Teknolojileri

- Sistem spesifik olmak
- Sınırlı widget kütüphanesi ?
- Move,resize vs. kaldırılması ?
- standalone, multiclient ?

- DirectFB (<http://www.directfb.org/>)
- Input aygıtları : klavye, ps/2 – seri mouse, joystick
- Image desteği : png, gif, jpeg
- libmpeg3 , libflash
- TrueType font desteği (FreeType2 ile)
- Standalone için ideal

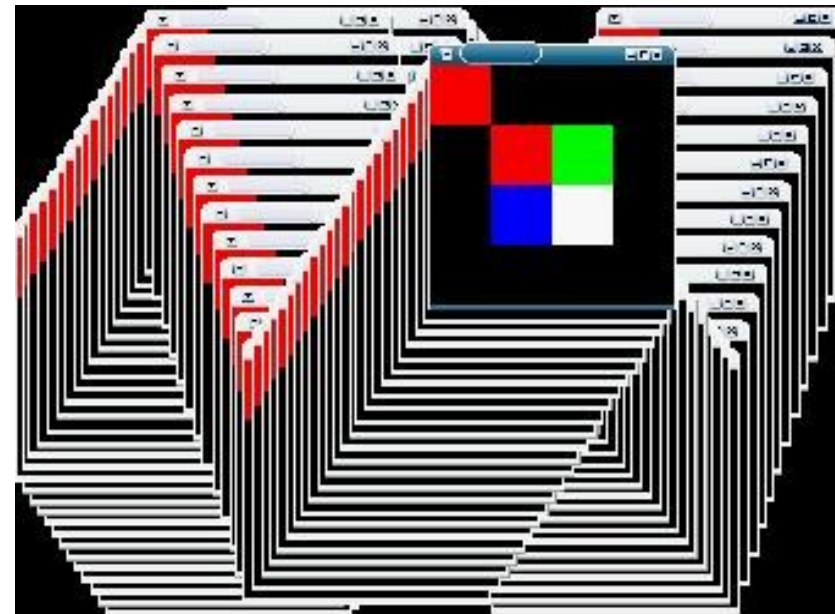


- PicoGUI (<http://picogui.org>)
- Compact, portable
- Widget ve layout motoru server'a gömülü
- Client/Server mimarisi (TCP/IP üzerinden)
- Small footprint



- Xynth (<http://gsulinux.org/~distch/?open=Projects&project=xynth>)

- Türk Malı :)
- Client / Server Model (UDS)
- DMA (Direct Memory Access to memory card)
- Temel windowing fonksiyonları (Move, 8-way resize)
- Static linked library : 120
- 1024X768x32 bit mode ve 128 tane client açıkken max. 3 Mb. Ram
- Portlanmış programlar : mplayer, mozilla, links
- Portlanmış library'ler : SDL, GTK 2.2
- GTKLib port



* Basic low-level graphics library for clients.

set_pixel

get_pixel

fill_box

get_box

put_box

copy_box

v_line

h_line

overlay functions;

set_pixel_o

get_pixel_o

fill_box_o

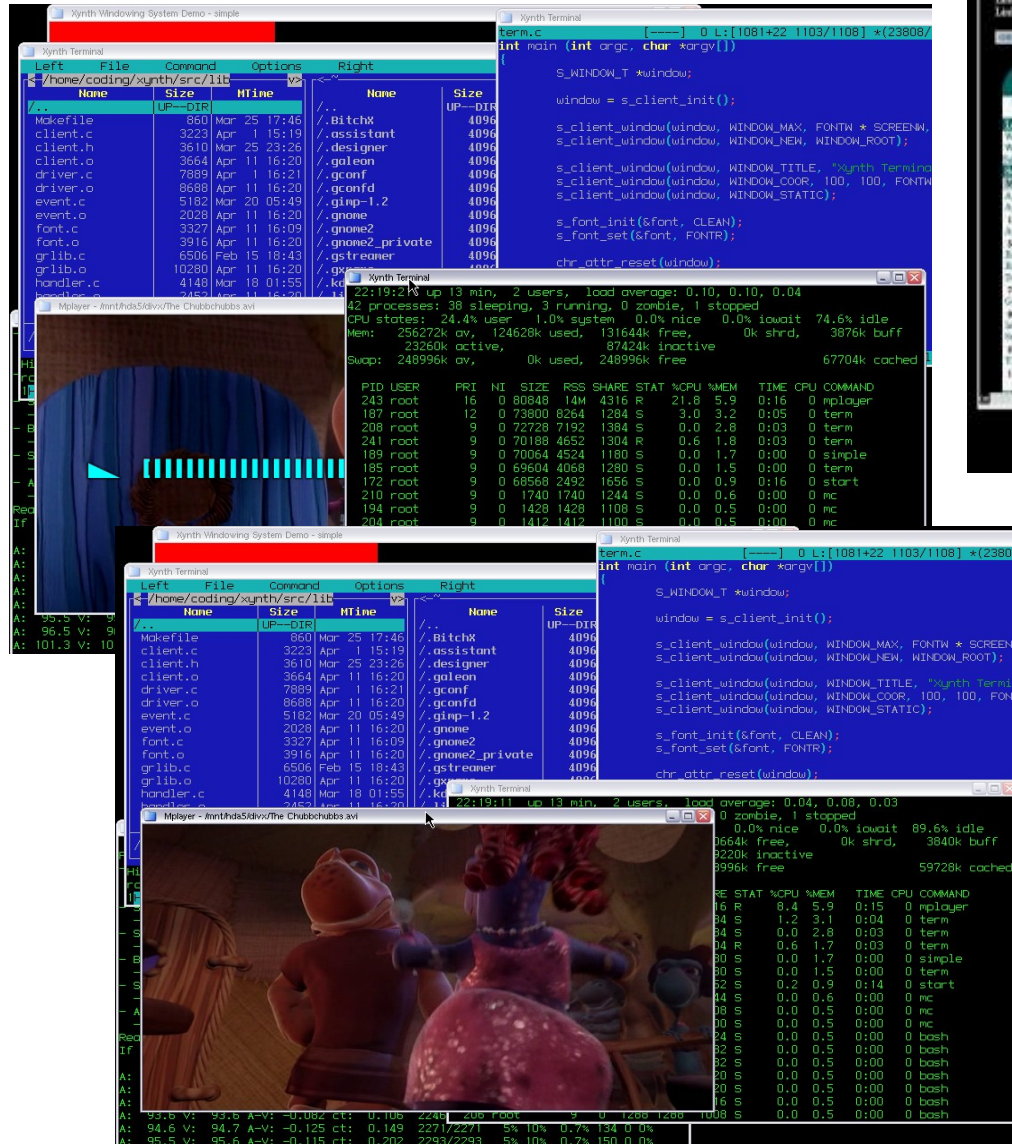
get_box_o

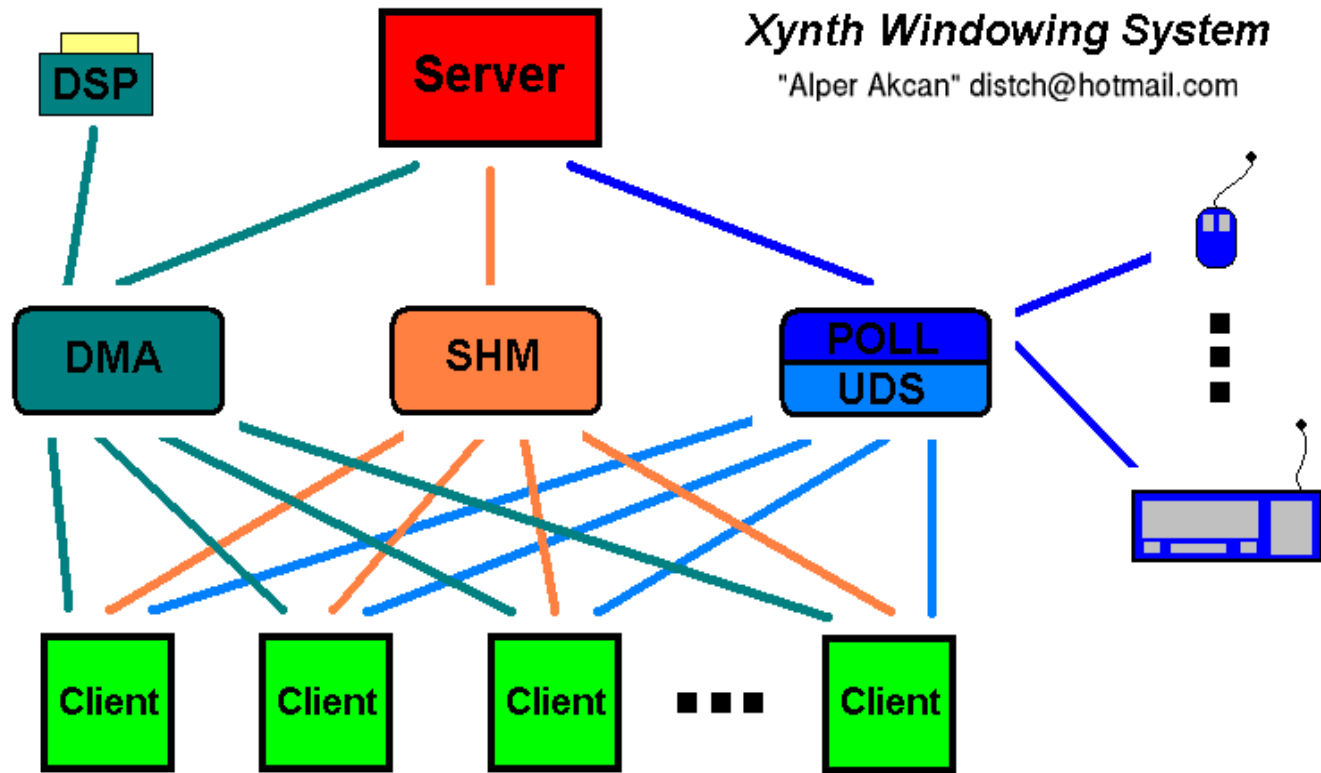
put_box_o

copy_box_o

vline_o

hline_o





DSP -> display

DMA -> vga buffer direct memory access

SHM -> shared memory, main screen bilgisi

UDS -> unix domain sockets

POLL -> serverin input devicelardan ve clientlardan input
recieve ettigi yer.

Contact Info :

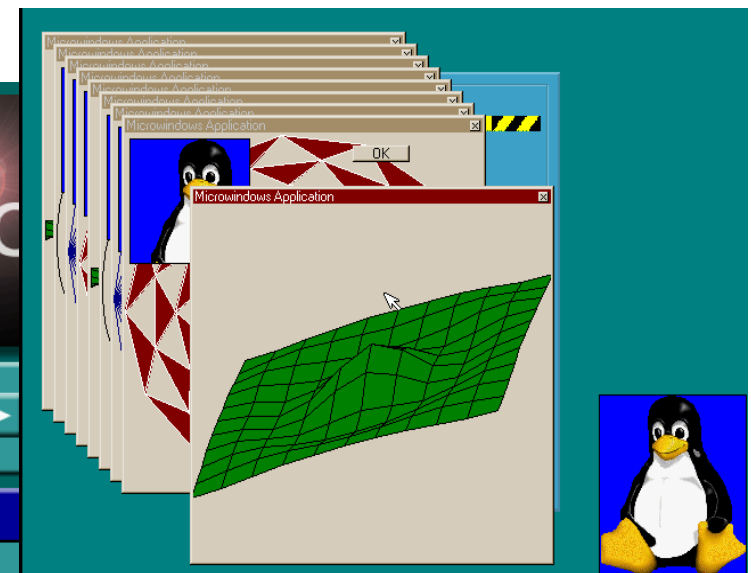
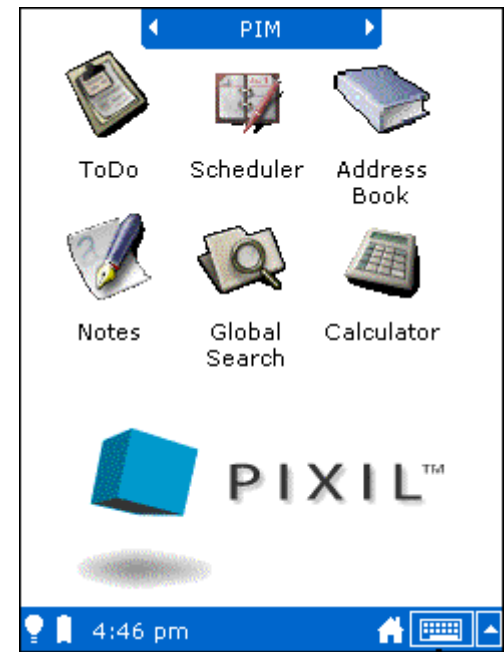
Alper Akçan

distch@hotmail.com

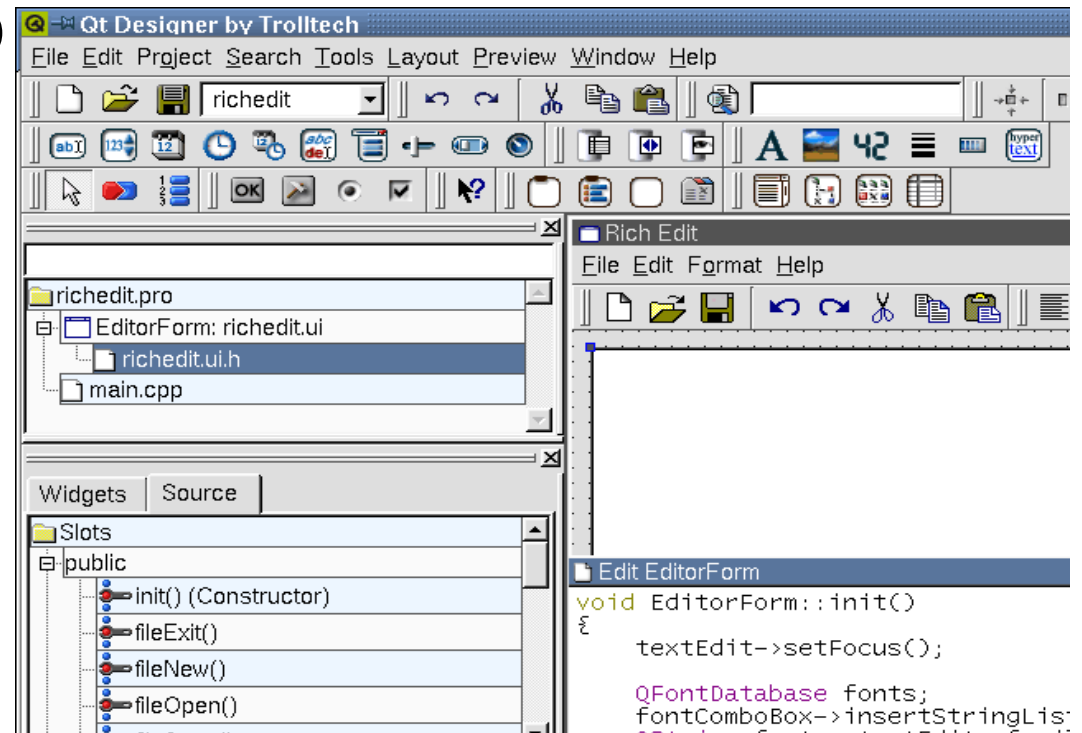
distch@gsu.linux.org.tr

- Microwindows (<http://microwindows.org/>)

- Microwindows API (WinCE GDI Standard)
- Nano-X API (UDS based Client/Server Model)
- Server ile link edilerek derlenebilir

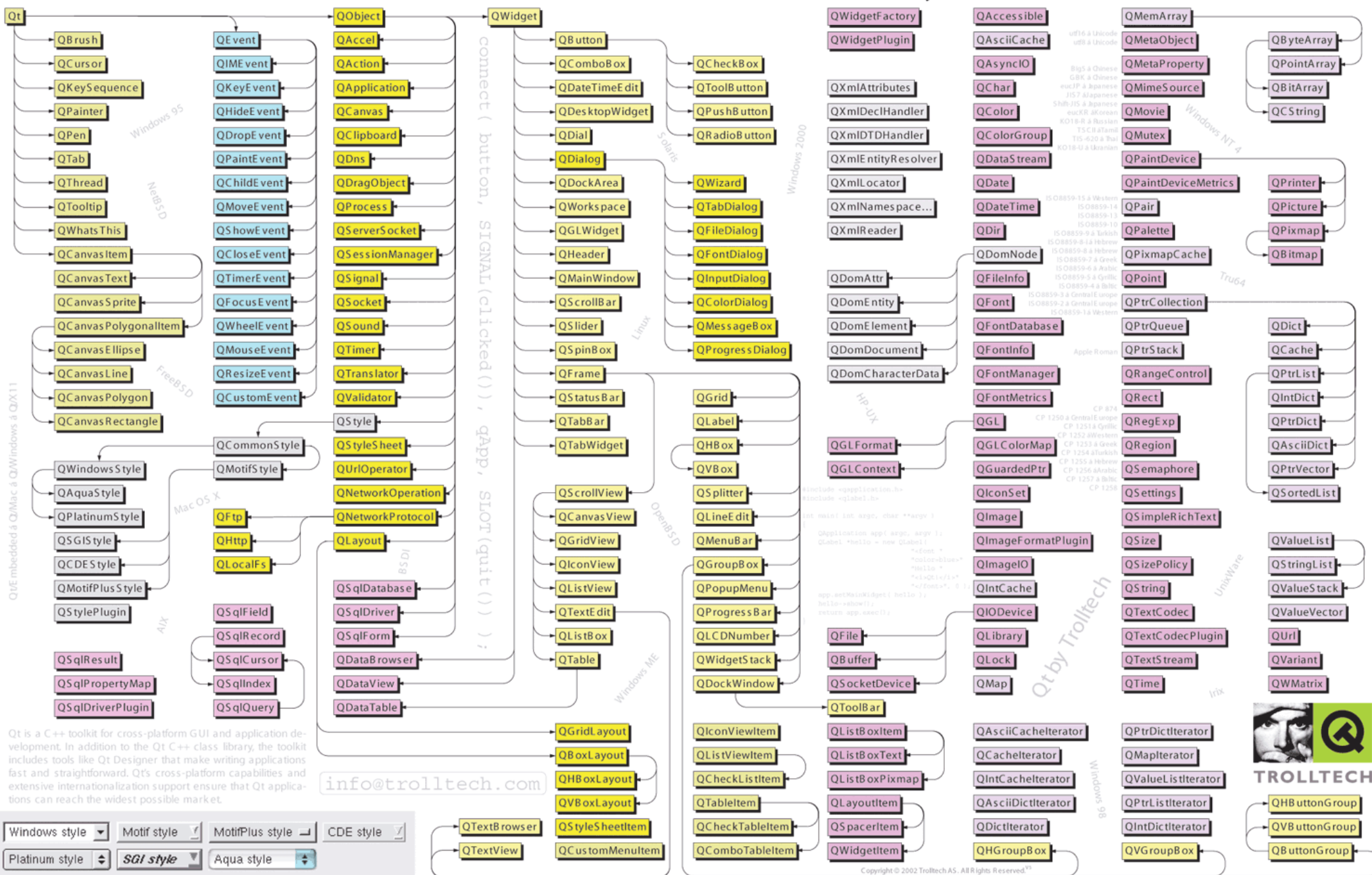


- QT/Embedded (<http://www.trolltech.com/products/embedded/index.html>)
- C++
- Xlib 'e ihtiyaç duymadan direk framebuffer üzerinde işlem yapar.
- 0.5 ile 2.5 Mb. arasında değişen library (compressed)
- QT-Designer (inline code editing)
- Database bağlantı
- Communicaton & Synchronization



- QT Class Hierarchy :

The Essential Qt 3.x Class Hierarchy



- Qtopia (<http://www.trolltech.com/products/qtopia/index.html>)

- QT/Embedded üzerine kurulu
- ~ 2.3 Mb. qt library
- ~ 6 Mb. full qt + applications (compressed)
- Özellikle PDA'ler için hazır platform
- Geniş Input yelpazesi
- Java Integration
- Bluetooth, IrDA communication (OBEX standard)
- Unicode tabanlı



Teşekkürler...

Referenslar :

www.tldp.org/HOWTO/Framebuffer-HOWTO.html

- linuxconsole.sourceforge.net/fbdev/HOWTO
- Linux Kernel Documentation

Derya Sezen

derya.sezen@eliar.com.tr

funky@gsu.linux.org.tr

thanx to Alper Akçan

distch@gsu.linux.org.tr

Lisans :

Software Foundation tarafından yayınlanmış bulunan GNU
Özgür Belgeleme Lisansının 1.2 ya da daha sonraki sürümünün koşullarına
bağlı kalarak kopyalayabilir, dağıtabilir ve/veya değiştirebilirsiniz.
Lisans'ın bir kopyasını <http://www.gnu.org/copyleft/gfdl.html> adresinde
bulabilirsiniz.

Bu belgedeki bilgilerin kullanımından doğacak sorumluluklar ve olası
zararlardan belge yazarı sorumlu tutulamaz. Bu belgedeki bilgileri uygulama
sorumluluğu uygulayana aittir.

Tüm telif hakları aksi özellikle belirtilmediği sürece sahibine aittir.

Belge içinde geçen herhangi bir terim, bir ticari isim ya da kuruma itibar
kazandırma olarak algılanmamalıdır. Bir ürün ya da markanın kullanılmış
olması ona onay verildiği anlamında görülmemelidir.