

Qt ile Bir Ağ Uygulaması

Aykut Aksoy

aykutaksoy@myway.com

sürüm 1.0

Özet

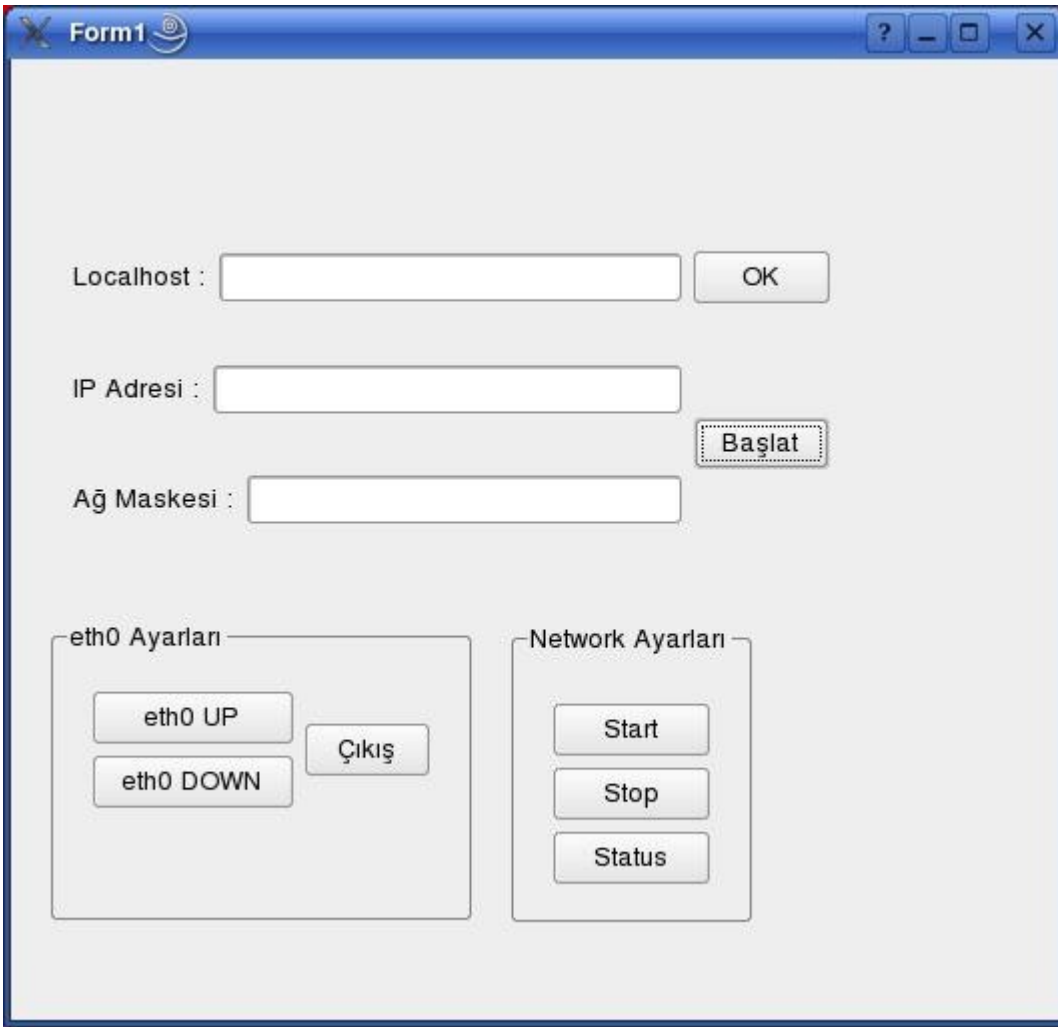
Bu belge Qt ile hazırlanmış bir ağ uygulamasını anlatır. C++ bilinmesi beklenir. Bu belgede "QProcess.h" kütüphanesine ağırlık verilmiştir. Bu proje "QProcess.h" kütüphanesini öğrenmek amacıyla yapılmıştır. Bu yüzden bir anlatı şeklindedir.

Ön Çalışma

Bu uygulamayı yazmadan önce bir ön çalışma yapmak ihtiyacı duydum. Sistemdeki,"ifconfig","ifconfig eth0","ifconfig lo","service network restart/stop/status" gibi komutların ne işe yaradığını bilmek gerekiyor. Ayrıca sistemin ağ ayarları ile ilgili dosyaların neler olduğu ve nerelerde tutulduğunu araştırdım.

Başlangıç

Yukarıdaki komutları araştırdıktan sonra, QTDesigner ile bir uygulama arayüzü oluşturmak gerekiyordu. Arayüzü kullanıcının bazı değerleri girebileceği ve ayarları değiştirebileceği şekilde tasarladım. Kullanıcı,sisteme ait ip adresi ve ağ maskesi gibi değerleri değiştirebiliyor. Ayrıca "eth0 Ayarları" ve "Network Ayarları" adlı iki buton table ile ilgili ayarlara ait butonları bir araya getirdim. Aşağıda, oluşturduğum arayüze ait bir resim bulunmaktadır:



Projenin Gelişimi ve Kaynak Kodu

Daha önce de bahsettiğim gibi uygulamanın neredeyse tamamını "QProcess.h" kütüphanesi kullanılarak yaptım. Bu kütüphaneyi kullanmam gerektiğini anlamam biraz zaman aldı. "QProcess.h" kütüphanesiyle, sistemde çalıştırılabilir olan uygulamaları başlatabiliyoruz. Bu uygulamalar komut, dosya veya program olabiliyor. Bu yüzden "QProcess.h" sınıfına ait fonksiyonların kullanımı çok önemli.

Proje kapsamında kullanıcı ip adresi, ağ maskesi gibi değerleri değiştirebilecek ve eth0 ayarlarını ve network start/stop/status gibi ayarları kontrol edebilecekti. "QProcess.h" kütüphanesini kullanarak, bu işleri gerçekleştiren sistem komutlarını Qt ile birlikte çalışabilecek hale getirebiliyoruz.

Butonlarla form arasındaki gerekli SIGNAL/SLOT ilişkilerini ayarlayıp, SLOT' lara uygun fonksiyonları tanımlıyoruz. "QProcess.h" 'dan bir pointer tanımladıktan sonra sınıfa ait "addArgument" ve "setArguments" gibi fonksiyonları kullanarak, istediğimiz komut veya programı çalıştırabiliyoruz. Sınıfın "addArgument(parametre)" fonksiyonunu kullanarak "ifconfig" komutunu çalıştırdım. Aşağıdaki kodda umarım demek istediklerim daha iyi anlaşılacaktır:

```
void Form1::local()
{ QString str=lineEdit5->text();
QProcess *p= new QProcess;
p= new QProcess( this );
p->addArgument("ifconfig");
p->addArgument("lo");
p->addArgument( str );
}
```

Burada "p" sınıfa ait bir nesneyi işaret ediyor. Önce lineEdit'in text()'ini "QString" sınıfından "str" adlı bir stringe atıyoruz.Bu işlemle lineEdit'e girdiğimiz ifadeyi "addArgument()" fonksiyonunun bir parametresi olarak alıyoruz."addArgument()" fonksiyonunun parametreleri string türündendir.Koddaki ilk parametre komutun kendisi, diğer parametreler, komuta ait parametrelerdir. Form1'e ait local() fonksiyonu, butona tıklatıldığında çalışacak olan SLOT'tur. Sonuçta yeni bir process start olur ve komut, butona tıklandığında çalışır hale gelir.Bu komutla, arayüzdeki satıra girceğimiz adres localhost olarak ayarlanır.

Buradaki sürecin(process) başlayıp başlamadığını kontrol etmek için aşağıdaki satırları koda ekledim:

```
if ( p->start() ) {
QMessageBox::information( 0,
tr("ok"),
tr("localhost aktif"),
tr("cikis") );}
else printf("acilmadi");
}
```

Görüldüğü gibi start() fonksiyonu sürecin başlayıp başlamadığını bildirir. Aynı yöntemi ip adresi ve ağ maskesi atamaları için de kullanıyoruz.

Uygulamanın eth0 ayarları kısmında eth0'ın iki durumunu kontrol ediyorum. Bunlar eth0 UP ve eth0 DOWN.Bunları da ilgili buton ve connection'ları ayarladıktan sonra, "ifconfig eth0 up/down" komutunun QProcess yardımıyla gerçekleştirmiyi yapıyoruz:

```
void Form1::ethup()
{
QProcess *p= new QProcess;
p=new QProcess(this);
p->addArgument("ifconfig");
p->addArgument("eth0");
p->addArgument("up");
if ( p->start() ) {
QMessageBox::information( 0,
tr("ok"),
tr("eth0 up"),
tr("cikis") );}
else printf("acilmadi");
}
```

Görüldüğü gibi aynı işlemi yapan bir kod olduğu için yukarıdakiyle oldukça benzer bir yapıda.Eth0 DOWN işlemini de aynı şekilde yapıyoruz.Zaten tüm projeyi bu yöntemle gerçekleştirdiğim söylenebilir.Bazı yerlerde dosyadan okuma kısımları dışında...

Bu işlemlerden sonra gelelim network servis ayarlarına.. Bilindiği gibi sistemde bu iş "service network start/stop/status" komutlarıyla gerçekleştiriliyor.İşte bu komutları QProcess.h ile gerçekleştiriyoruz.

```
void Form1::network()
{
QProcess *p= new QProcess;
p=new QProcess(this);
p->addArgument("service");
p->addArgument("network");
p->addArgument("start");
if ( p->start() ) {
QMessageBox::information( 0,
tr("ok"),
tr("network restart edildi"),
```

```
tr("cikis" ) ;}
else printf("acilmadı");
}
```

Stop komutunu da aynı şekilde gerçekleştirdikten sonra status komutuna sıra geliyor.Yalnız burada bir farklılık var. "service network status" komutunun çıktısı ekrana network'un o anki durumu hakkında bilgi verir.Dolayısıyla bu komutun çıktısını bir dosyaya yazdırıp, QProcess ile o dosyadan okutmamız gerekiyor.Aşağıdaki kodu incelersek:

```
void Form1::durum()
{ FILE *ptr;int boyut;
QProcess *p= new QProcess;
p=new QProcess(this);
p->addArgument("service");
p->addArgument("network");
p->addArgument("status");
p->addArgument(">/root/Desktop/bilgi.txt");
ptr=fopen("/root/Desktop/bilgi.txt","r");
fseek(ptr,0,SEEK_END);
boyut=ftell(ptr);
fseek(ptr,0,SEEK_SET);
char *buf=(char*)malloc(boyut);
fread(buf,1,boyut,ptr);
if ( p->start() ) {
QMessageBox::information( 0,
tr("ok"),
tr(buf),
tr("cikis" ) ;}
else printf("acilmadı");
}
```

Bu kodla "service network status" komutunun çıktısını "/root/Desktop" altındaki "bilgi.txt" dosyasına yazdırıyoruz. Daha sonra bu dosyayı read modunda açıp, bellekten dosya boyutu kadar yer ayırıyoruz.Bu dosyanın içeriğini de fread() ile okuyor ve butona tıklandığında bir information box ile ekrana çıkmasını sağlıyoruz.Aslında başında çalıştırdığımız komut masaüstündeki dosyaya yazdırmayı da içeriyor("service network status>/root/Desktop/bilgi.txt").Bu sadece alternatif bir yol tabii ki, dileyen daha farklı yöntemlerle gerçekleştirebilir...

Sonuç olarak bir toparlamak gerekirse "form1.ui.h" ve "main.cpp" dosyalarının içeriği aşağıdaki gibi olmalıdır:

Form1.ui.h

```
#include qprocess.h
#include qstringlist.h
#include stdlib.h
#include qtextview.h
#include qmessagebox.h
#include stdio.h
void Form1::init(){
}
void Form1::local()
{ QString str=lineEdit5->text();
QProcess *p= new QProcess;
p= new QProcess( this );
p->addArgument("ifconfig");
```

```

p->addArgument("lo");
p->addArgument( str );
if ( p->start() ) {
QMessageBox::information( 0,
tr("ok"),
tr("localhost aktif"),
tr("cikis" ) );}
else printf("acilmadı");
}
void Form1::baslat()
{QString str=line1->text();
QString str2=line2->text();
QProcess *p= new QProcess;
p=new QProcess(this);
p->addArgument("ifconfig");
p->addArgument("eth0");
p->addArgument(str);
p->addArgument("netmask");
p->addArgument(str2);
if ( p->start() ) {
QMessageBox::information( 0,
tr("ok"),
tr("baslatildi"),
tr("cikis" ) );}
else printf("acilmadı");
}
void Form1::ethup()
{
QProcess *p= new QProcess;
p=new QProcess(this);
p->addArgument("ifconfig")
; p->addArgument("eth0");
p->addArgument("up");
if ( p->start() ) {
QMessageBox::information( 0,
tr("ok"),
tr("eth0 up"),
tr("cikis" ) );}
else printf("acilmadı");
}
void Form1::ethdown()
{
Process *p= new QProcess;
p=new QProcess(this);
p->addArgument("ifconfig");
p->addArgument("eth0");
p->addArgument("down");
if ( p->start() ) {
QMessageBox::information( 0,
tr("ok"),
tr("eth0 down"),
tr("cikis" ) );}
else printf("acilmadı");
}
void Form1::network()
{
QProcess *p= new QProcess;

```

```

p=new QProcess(this);
p->addArgument("service");
p->addArgument("network");
p->addArgument("start");
if ( p->start() ) {
QMessageBox::information( 0,
tr("ok"),
tr("network restart edildi"),
tr("cikis") );}
else printf("acilmadı");
}
void Form1::stop()
{
QProcess *p= new QProcess;
p=new QProcess(this);
p->addArgument("service");
p->addArgument("network");
p->addArgument("stop");
if ( p->start() ) {
QMessageBox::information( 0,
tr("ok"),
tr("servis durduruldu"),
tr("cikis") );}
else printf("acilmadı");
}
void Form1::durum()
{ FILE *ptr;int boyut;
QProcess *p= new QProcess;
p=new QProcess(this);
p->addArgument("service");
p->addArgument("network");
p->addArgument("status");
p->addArgument(">/root/Desktop/bilgi.txt");
ptr=fopen("/root/Desktop/bilgi.txt","r");
fseek(ptr,0,SEEK_END);
boyut=ftell(ptr);
fseek(ptr,0,SEEK_SET);
char *buf=(char*)malloc(boyut);
fread(buf,1,boyut,ptr);
if ( p->start() ) {
QMessageBox::information( 0,
tr("ok"),
tr(buf),
tr("cikis") );}
else printf("acilmadı");
}

```

Main.cpp

```

#include qapplication.h
#include "form1.h"
int main( int argc, char ** argv )
{
QApplication a( argc, argv );
Form1 w;
w.show();

```

```
a.connect( &a, SIGNAL( lastWindowClosed() ), &a, SLOT( quit() ) );  
return a.exec();  
}
```

Son Söz

Görüldüğü gibi program gayet basit.Umarım QT Türkiye'ye bir katkı olmuştur.Herkese teşekkürler.

Telif Hakkı ve Lisans

Bu belgenin, Qt ile Bir Ağ Uygulaması , 1.0 sürümünün telif hakkı © 2006 Aykut AKSOY'a aittir. Bu belgeyi, Free Software Foundation tarafından yayınlanmış bulunan GNU Özgür Belgeleme Lisansının 1.1 ya da daha sonraki sürümünün koşullarına bağlı kalarak kopyalayabilir, dağıtabilir ve/veya değiştirebilirsiniz. Bu Lisansın bir kopyasını <http://www.gnu.org/copyleft/fdl.html> adresinde bulabilirsiniz.

Linux, Linus Torvalds adına kayıtlı bir ticarî isimdir.
Qt, TrollTech adına kayıtlı bir ticari isimdir.

Feragatname

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 ticarî isim ya da kuruma itibar kazandırma olarak algılanmamalıdır. Bir ürün ya da markanın kullanılmış olması ona onay verildiği anlamında görülmemelidir.