

# Linux ile Programlama

## Linux Kullanıcıları Derneği

<http://www.linux.org.tr>   <http://www.lkd.org.tr>

**Barış Metin**  
**baris@uludag.org.tr**

**Serdar Köylü**  
**serdar@uludag.org.tr**

# Özgürsünüz!

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

# İçerik

- Programlama
- Linux ortamında programcı
- Dezavantajlar
- Diller/Ortamlar
- Kütüphaneler
- Yardımcı araçlar
- Özgür yazılım projelerine ilk adımlar

# Programlama

- Bilgisayara işi nasıl yapacağını anlatmaktır.
- Bilgisayar aptaldır! Bir işi “aptal”a anlatırsanız, kendiniz de daha iyi anlarsınız ;). (iş kavrayışınızdaki muğlaklığı, ortadan kaldırırsınız)
- Tembelliktir!
- Eğlencedir!

# Linux: Programcının Evi

- Bilgi birikimi: Belgeler ve kaynak kod...
- Tam donanımlı araçlar.
- Esnek yapı.
- Binlerce programcı ortağınız/arkadaşınız.



**Fotoğraf: Rahat bir yazılım geliştirme ortamı.**

# Eksiler (!)

- Dağınık bileşenler
- Hızlı gelişen/değişen projeler
- Yetersiz dökümantasyon

# Dağınık Bileşenler

- Küçük, becerikli bileşenler çok.
- Bileşenleri birleştirmek zevkli, fakat ilk adımı atmayı zorlaştırıyor.
- Programlama ortamının gücünü de bu bileşen bolluğu sağlıyor.
- Bütünleşik geliştirme ortamları mevcut.

# Gelişen Ortam

- Özgür yazılım geliştiricileri çalışan.
- Ortam yaşıyor, sürekli gelişiyor.
- Geriye uyumluluk büyük oranda korunuyor.
- Fakat son teknolojiyi yakalamak bilgi, deneyim ve çaba istiyor.



# Programcı Belgeleri

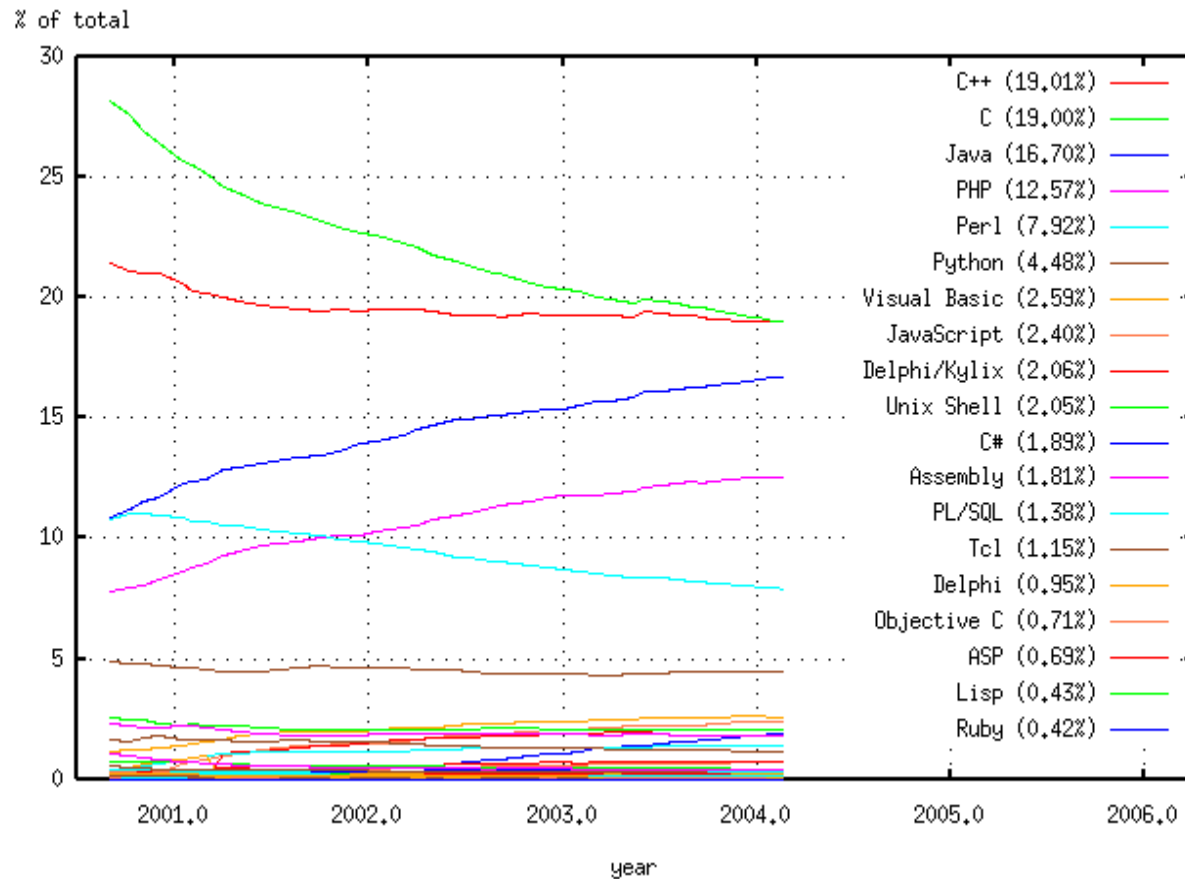
- Programcıya yol gösterici belgeler, örnekler, vs. çok.
- Çalışma ortamları, kütüphaneler, genellikle çok iyi belgeleniyor.
- Fakat bazı durumlarda belgeler geliştirmenin hızını yakalayamıyor.
- Gelişimi takip etmek için en iyi yol geliştirmenin içinde olmak (e-posta listeleri, CVS).
- Kısaca: Ekmek yoksa pasta yiyin! (Döküman yoksa kodu okuyun)

# Neler var?

**Php**  
**Python**  
**Ruby**  
**Aleph**  
**Java**  
**Mono/C#**  
**C++**  
**Perl**  
**Mathlab**  
**C**  
**Bash**  
**Pascal**  
**Fortran**  
**Octave**  
**Haskell**  
**Lisp**  
**Awk**  
**Ada**  
**Ocaml**  
**Asm**  
**Pike**  
**SmallTalk**  
**Prolog**  
**Kylix**  
**TCL**

# Hangi Dil?

C++	13.529
C	13.446
Java	12.096
PHP	9.047
Perl	5.553
Python	3.256
Visual Basic	1.842
JavaScript	1.743
Delphi/Kylix	1.475
UnixShell	1.472



# GNU Compiler Collection

- “C”den daha fazlasını derler

C, C++, Objective C, GCC (çeşitli C “lehçeleri”), Fortran, Pascal, Modula-3, Ada . . .

gcc : C stili bağlama ile derler

g++: C++ stili bağlama ile derler

- GCC modülerdir

Önişleme (cpp)

gcc -E

Derleme (gcc)

gcc -S

Object code (gas)

gcc -c

Bağlayıcı (ld)

# Hata Ayıklamak

- GDB üst seviye bir hata ayıklayıcıdır (debugger).

```
/*
 a buggy code to debug
 */

#include <unistd.h>
#include <stdlib.h>

#define BIGNUM 5000

void build_index(int ary[]);

int main(void)
{
    int intary[100];
    build_index(intary);
    exit(EXIT_SUCCESS);
}

void build_index(int ary[])
{
    int i;
    for(i = 0; i < BIGNUM; ++i)
        ary[i] = i;
}
```

```
baris@evliya:~$ gdb -q a.out
(gdb) run
Starting program: /home/baris/a.out
```

```
Program received signal SIGSEGV, Segmentation fault.
0x080483b9 in build_index (ary=0xbffff840) at a.c:23
23         ary[i] = i;
(gdb) backtrace
#0 0x080483b9 in build_index (ary=0xbffff840) at a.c:23
#1 0x08048385 in main () at a.c:15
(gdb) list
15     build_index(intary);
16     exit(EXIT_SUCCESS);
17 }
18
19 void build_index(int ary[])
20 {
21     int i;
22     for(i = 0; i < BIGNUM; ++i)
23         ary[i] = i;
24 }
(gdb) print i
$1 = 496
(gdb)
```

# Kütüphaneler

- Programcı için gerekli altyapı büyük oranda kütüphaneler ile sağlanır.

Temel Libc API

Yardımcı Kütüphaneler

Ağ haberleşmesi, Veri saklama, Text Arabirim, Grafik Arabirim,  
Alt seviye donanım, ...

libxml2, libxmlrpc, libcgi, libpam-\*, libldap2, gettext, glib, libpq,  
libwww, libace, libapr, libdb, libdiscover, gtk/gnome, qt/kde,  
libsnmp, libssl, libusb, newt, ncurses, AALib, libtermcap,  
libcurl...

# Temel libc API

**POSIX Uyumlu Sistem çağrıları..**

**syscall, ioctl, stat, fstat..**

**BSD & SVRv4 Uyumlu Sistem çağrıları..**

**memcpy, lstat, bcopy**

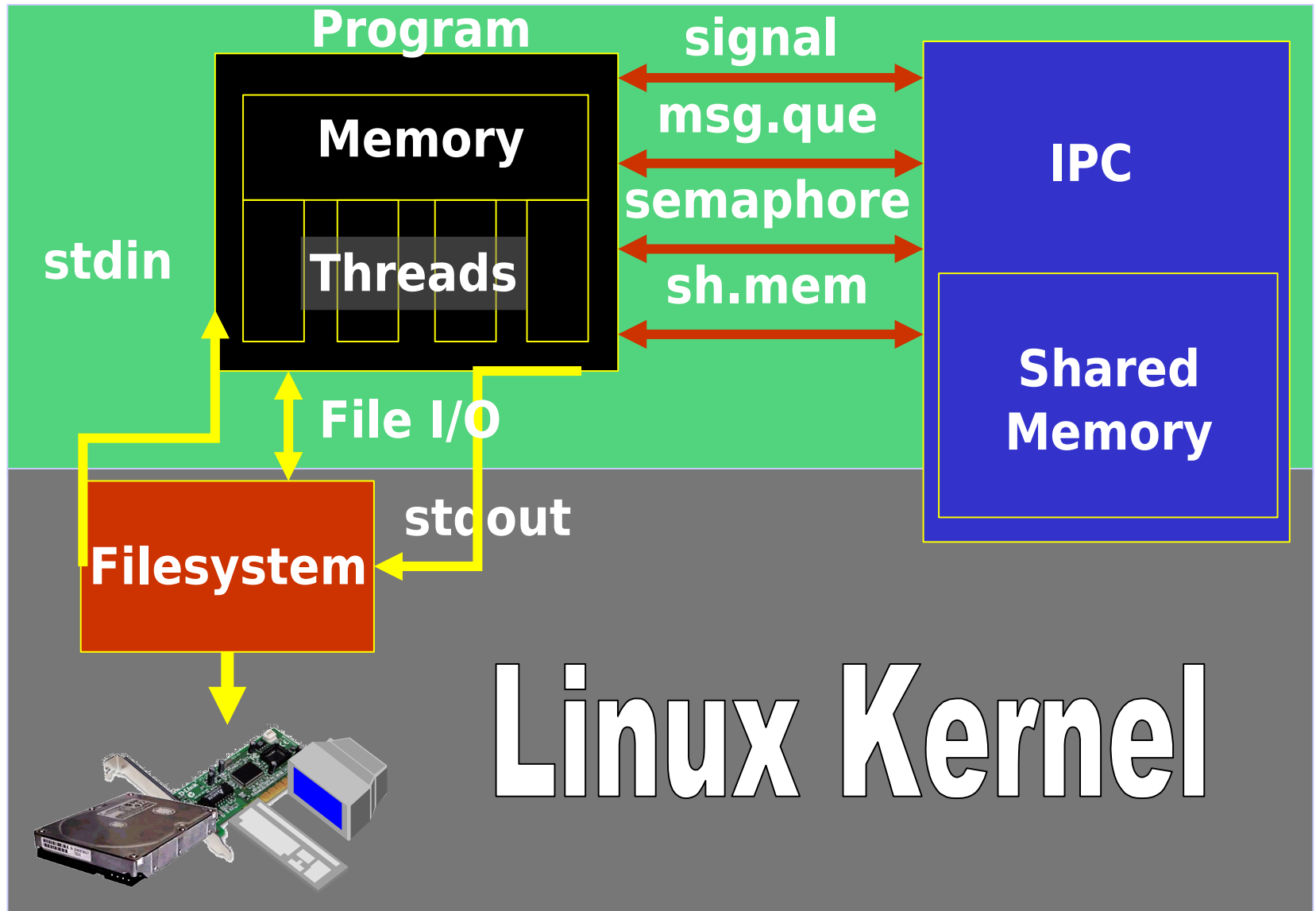
**ANSI-C Uyumlu Sistem çağrıları..**

**Calloc, printf, sprintf, fopen, readdir, system..**

**Linux'a özel sistem çağrıları..**

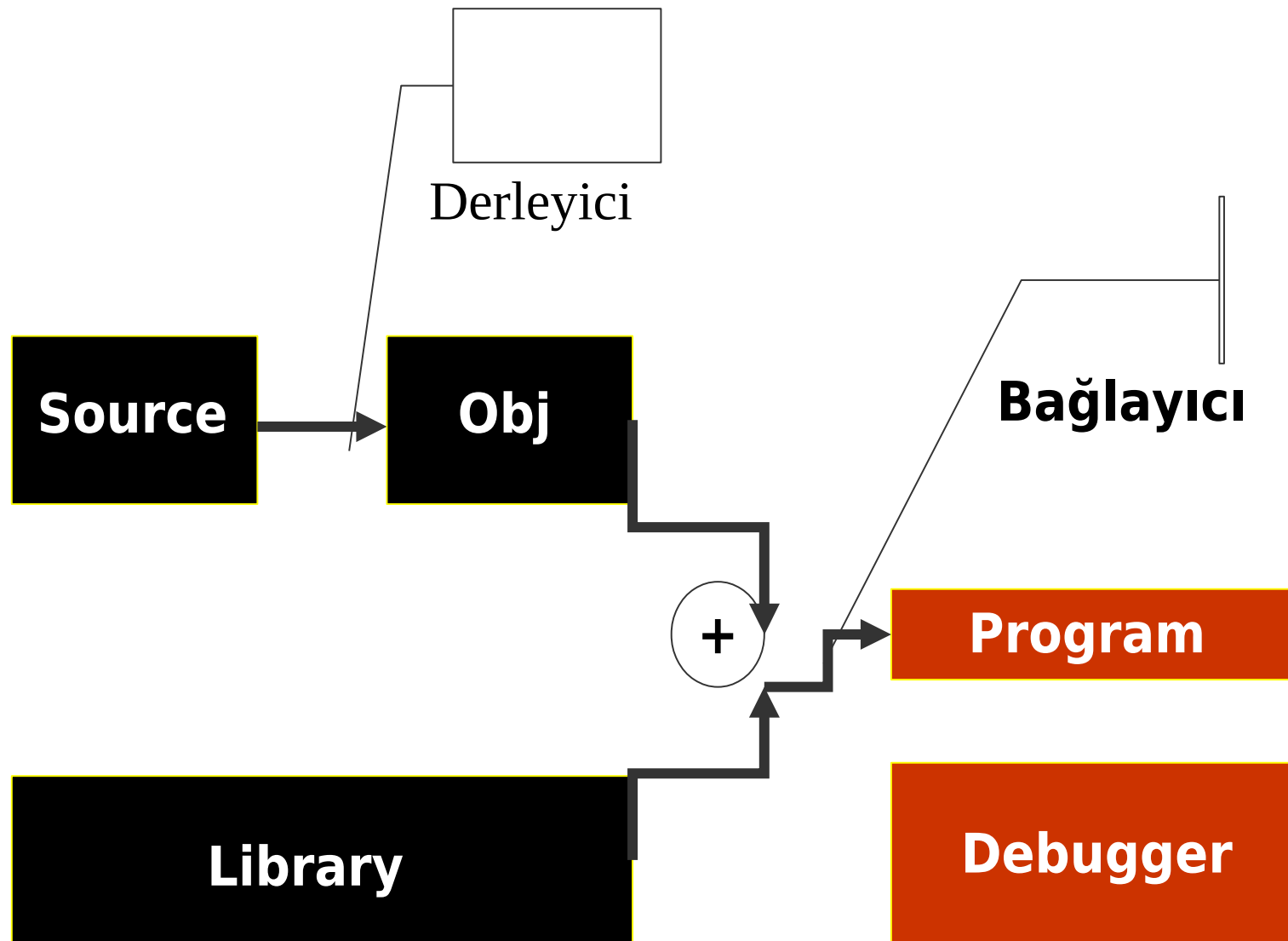
**Sound, apm vs. çağrıları..**

# IPC: Interprocess Communication



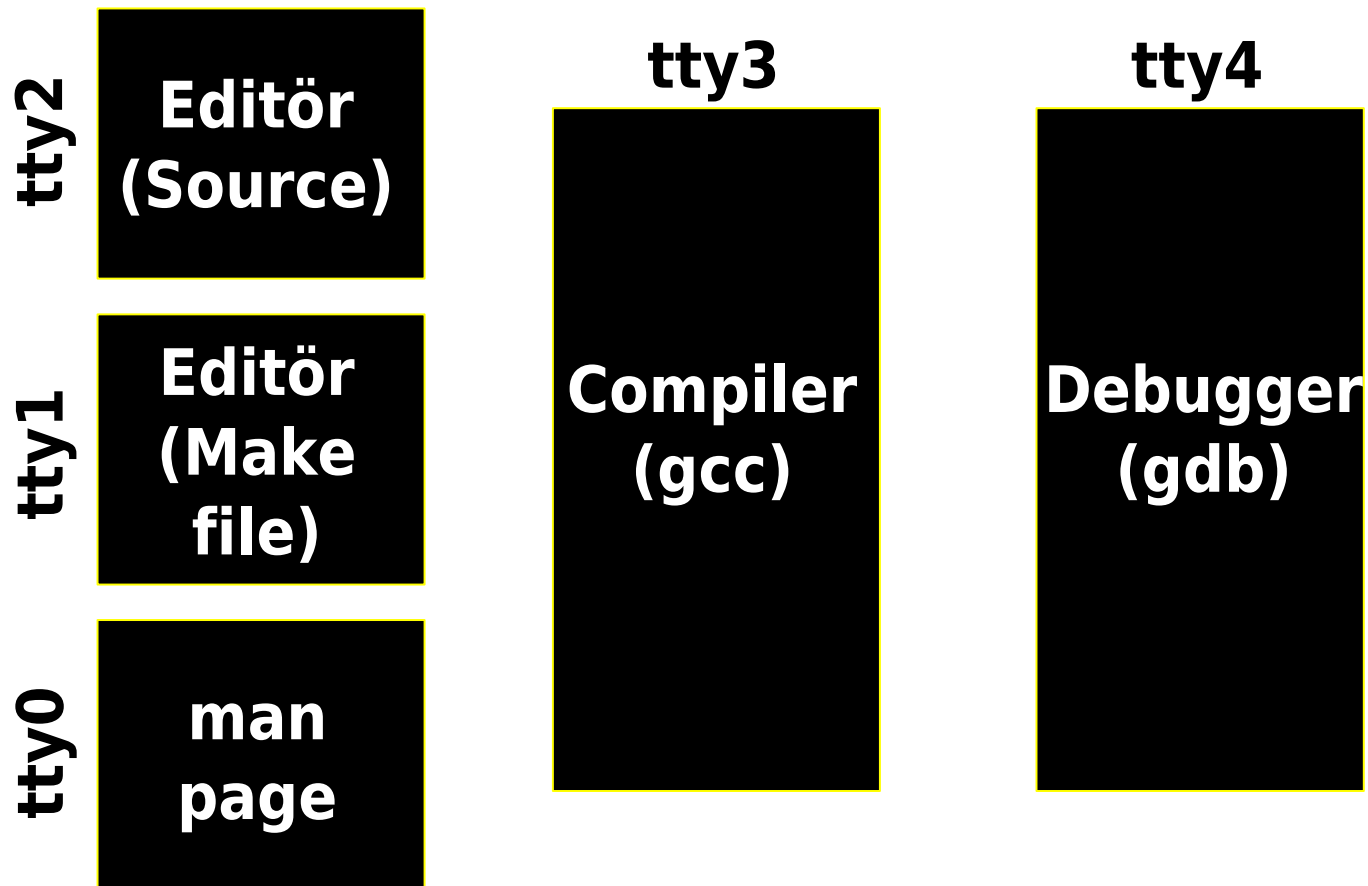


# Bir Programın Hayatı

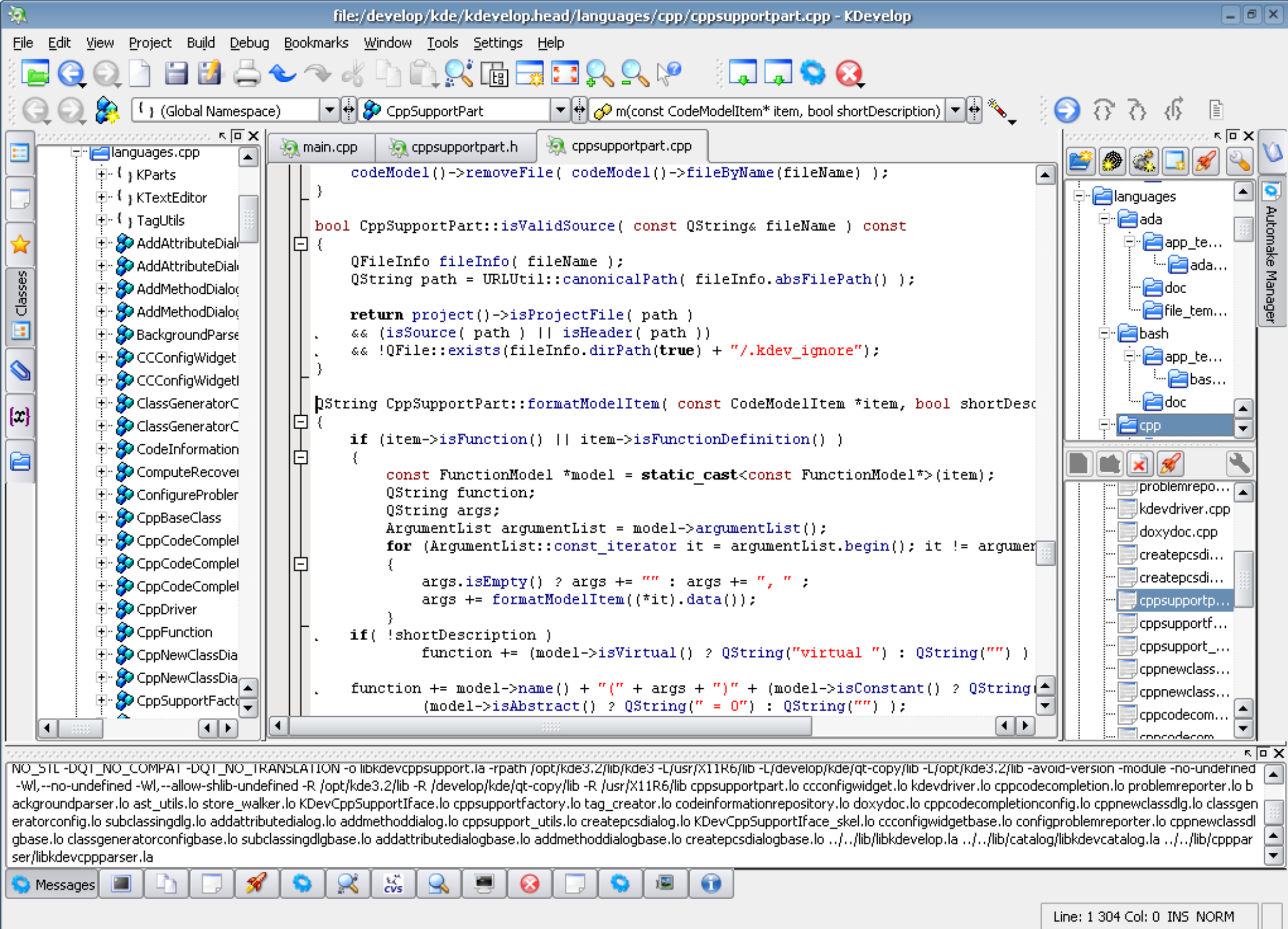


# UDE

## Unintegrated Development Environment



**Delikanlı Programcının Geliştirme Ortamı ;)**





```
konst@morgue: /home/konst
MOTOR 1.0.0 Program: artonline File: artonline.cc 0: 0
#include "artonline.h"

artonline::artonline(): tmp1(0) {
    pars
    valu
    html
}
artonlin
ppar
pars
list
html
dele
}

bool art
bool

try {
    ora = new GOracleInterface();
    ora->connect("art", "art", 0);
}

CVS: looking for modified files
```

# Yardımcı Araçlar

- **Sürüm Yönetimi:** RCS, CVS, Subversion
- **Hata takip:** Bugzilla, Gnats, debbugs
- **Program gözatıcı:** cbrowser, cscope, exuberant-ctags, cflows
- **Program kontrolcü:** valgrind, splint, pscan
- **Derleme:** make, ccache, autotools
- **Çağrı takip:** strace, ltrace
- **Dökümantasyon:** doxygen, cxref

# Nasıl katılacağız?

- **Dinleyin:** e-posta listeleri, irc kanalları, sunumlar, toplantılar
- **İnceleyin:** Topluluğu inceleyin, yaşam şartlarını öğrenin
- **Okuyun:** Belgeleri ve kaynak kodları okuyun.
- **Katlı verin:** Yeterince bilgi sahibi olduğunuzu düşündüğünüzde katkınızı ortaya koyun.

# Nasıl Katılacağız? (2)

- Proje üzerindeki tartışmaları araştırın.
- Proje (kaynak kod) organizasyonunu inceleyin: Dizin yapısı proje hakkında pek çok bilgi verebilir.
- “Nasıl yapmışlar” sorusunun yanıtı çoğu zaman kaynak koddan geçer.
- Programı derleyin, çalıştırın: Derleyici uyarıları, sistem çağrılarının analizi, oluşturulan çalıştırılabilir kodu incelemek pek çok fikir verebilir.
- Büyük projeleri yalnızca (küçük) bir bölümünü inceleyerek düzenleyebilirsiniz.
- Yazım stillerini dikkate alın.



# Nasıl Katılacağız? (3)

- Özgür yazılımlarda kimsenin “sorumlu” olmadığını bilin. Kimseyi yaptıklarından/yapmadıklarından dolayı suçlamayın.
- “X özelliği ne zaman eklenecek?” sorusu çoğu zaman yanıtsız kalabilir. Sorunun cevabını kendiniz vermeye çalışın
- Topluluk hakkında sürekli bilgi toplayın. Güncel haber sitelerini takip edin, topluluğun bir üyesi olun.
- Diğerleri ile sürekli iletişim içinde olun. Tek başınıza işe başlamayın.

# Nasıl Katılacağız? (4)

- Bulduğunuz hataları bildirin ve takip edin.
- Bildirilmiş hataları inceleyin.
- Düşüncelerinizi paylaşın
- Bir hatayı gözünüze kestirin ve kolları sıvayın.
- Size saldırılacak! Bunu bilin ve kabuğunuzu kalınlaştırın.
- Topluluğu inceledikten sonra kimlerden yapıcı eleştiriler alacağınızı öğreneceksiniz, önemseyin.
- Tüm eleştirileri de dikkate alın.
- Herkes gibi, eğlenin :)

# Biz de eğlenelim!

## Sohbet & Sorular