



FreeBSD'de C Programlama Dili ile Yazılım Geliştirme Teknikleri

Cihan Kömeçoğlu

EnderUNIX Yazılım Geliştirme Takımı Üyesi

cihan@enderunix.org

<http://www.enderunix.org/cihan/>

Sunum İeriđi

- Gcc ile c programı derleme
- Yama Oluřturma
- Gdb ile hata ayıklama
- Makefile oluřturma



GCC ile Derleme

- Tek bir kaynak dosyayı derlemek için

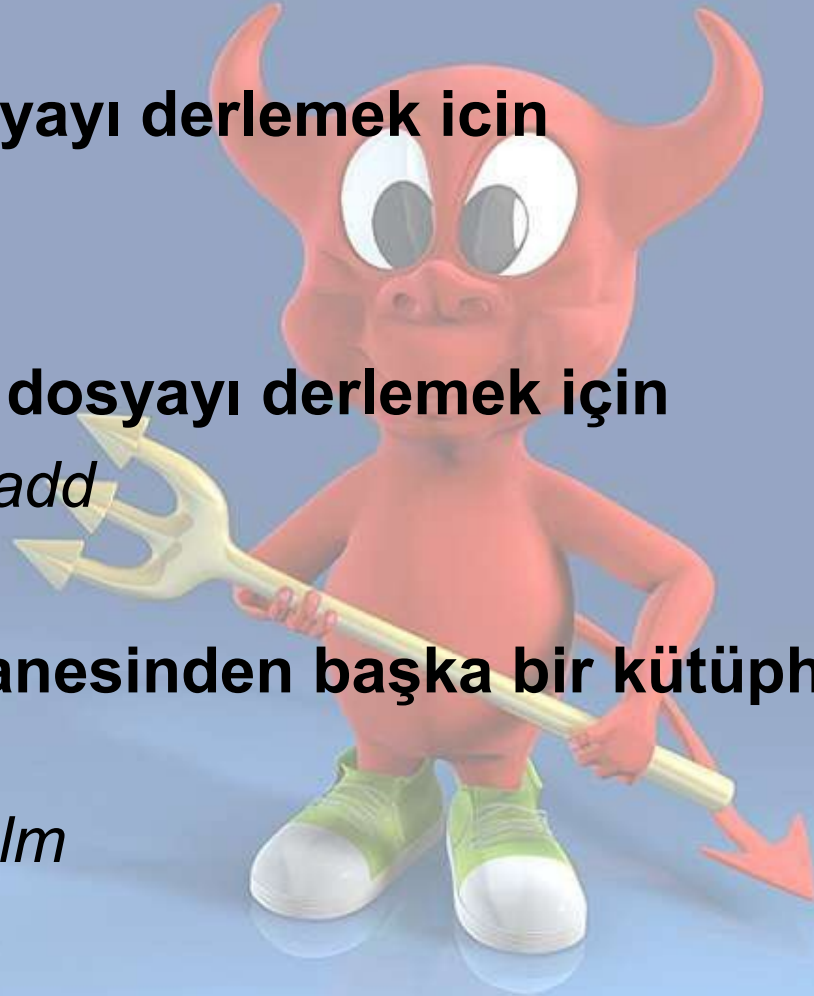
\$gcc ilk.c -o ilk

- Birden çok kaynak dosyayı derlemek için

\$gcc main.c func.c -o add

- Standart C kütüphanesinden başka bir kütüphane kullanıldığında

\$gcc math.c -o math -lm

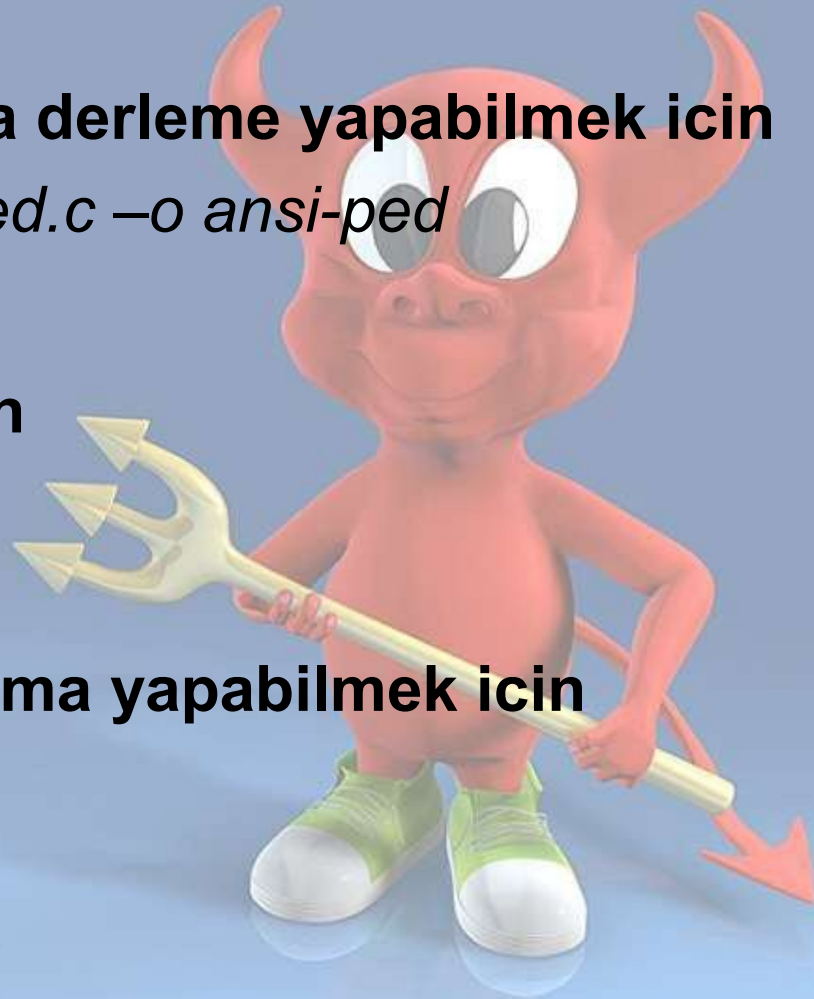


GCC ile Derleme

- **Kutuphane standart bir dizinde degilse**
\$gcc mysql.c -o mysql -L/usr/local/lib -lmysqlclient
- **Baslik dosyasi standart bir dizinde degilse**
\$gcc baslik.c -o baslik -I/usr/local/include
- **Derleme esnasindaki bütün uyarıları göstermek için**
\$gcc -Wall wall.c -o wall
- **Ansi C standardında derleme yapabilmek için**
\$gcc -ansi ansi-ped.c -o ansi-ped

GCC ile Derleme

- **ISO C standardında derleme yapabilmek için**
\$gcc -pedantic ansi-ped.c -o ansi-ped
- **Static derlemek için**
\$gcc -static ilk.c -o ilk
- **GDB ile hata ayıklama yapabilmek için**
\$gcc -g gdb.c -o gdb



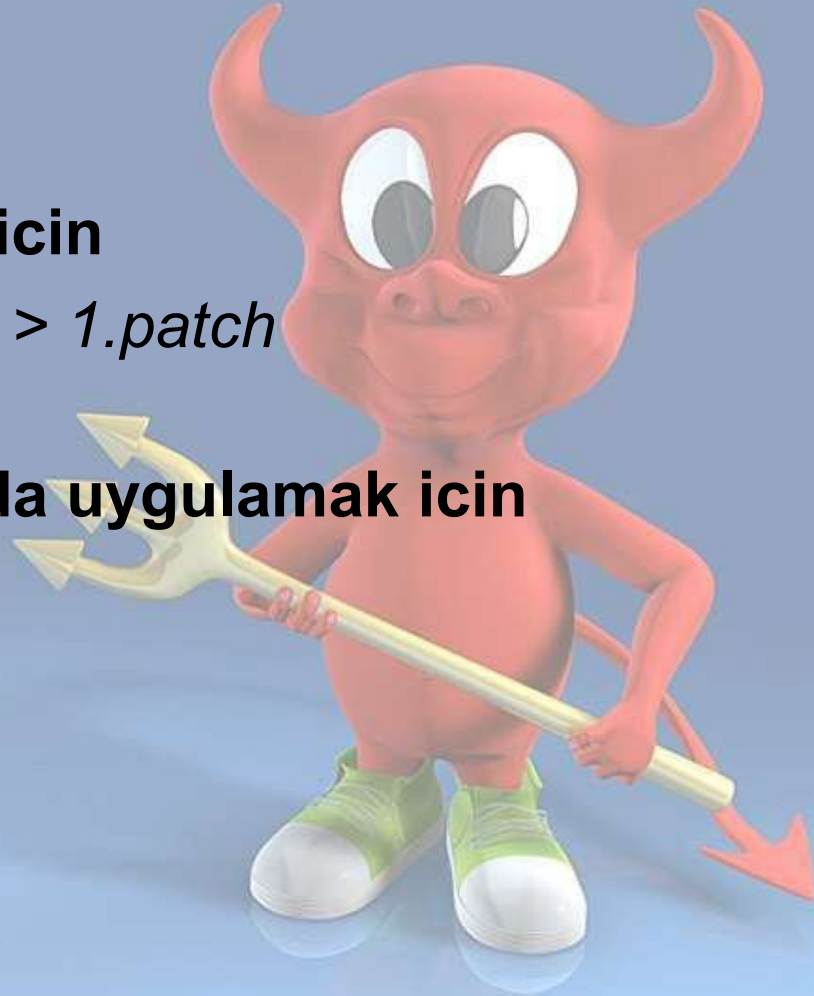
Yama Oluřturma

- Yama oluřturmak icin

diff -ruN func.c patch.c > 1.patch

- Yamayi kaynak koda uygulamak icin

patch < 1.patch



GDB ile Hata Ayıklama

- **Gdb ile hata ayıklamayı başlatmak için**
\$gdb program
- **Programı başlatmak için “run” komutu girilir.**
- **breakpoint koymak için**
break fonksiyon_adi
break satir_numarasi



GDB ile Hata Ayıklama

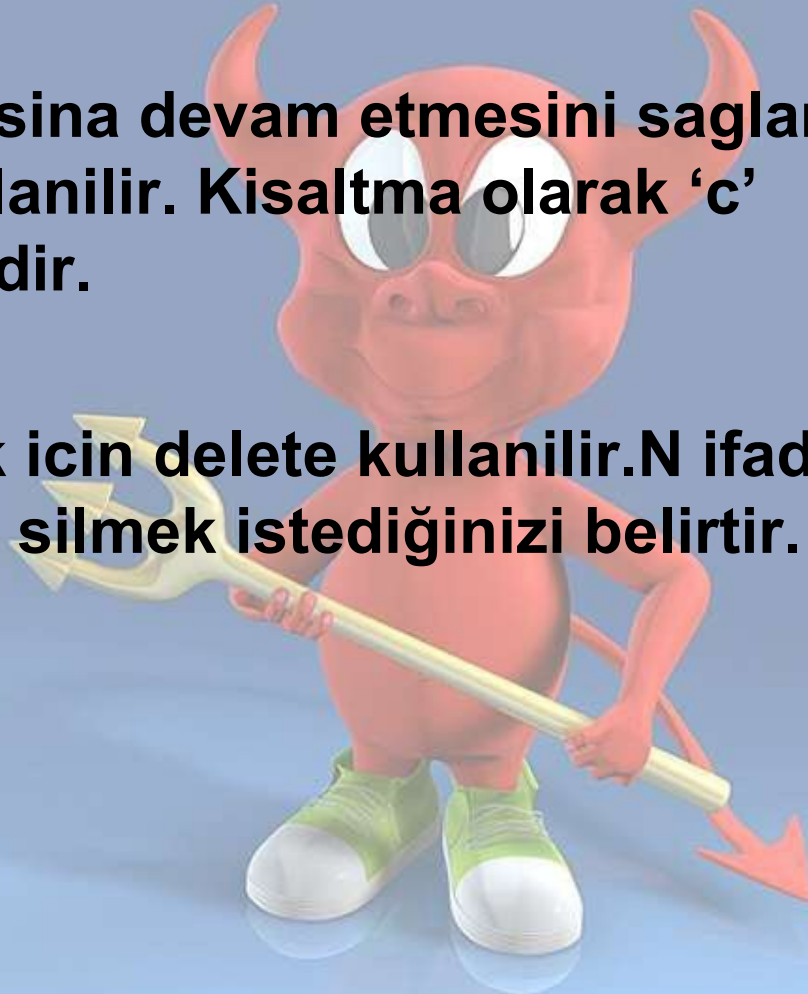
- **next, step bir sonraki satira ilerlemek icin kullanilir. Next icin 'n' , step icin 's' kullanilmasi yeterlidir.**
- **Degiskenin icerigini yazdirmek icin print kullanilir.**
print degisken_adi



GDB ile Hata Ayıklama

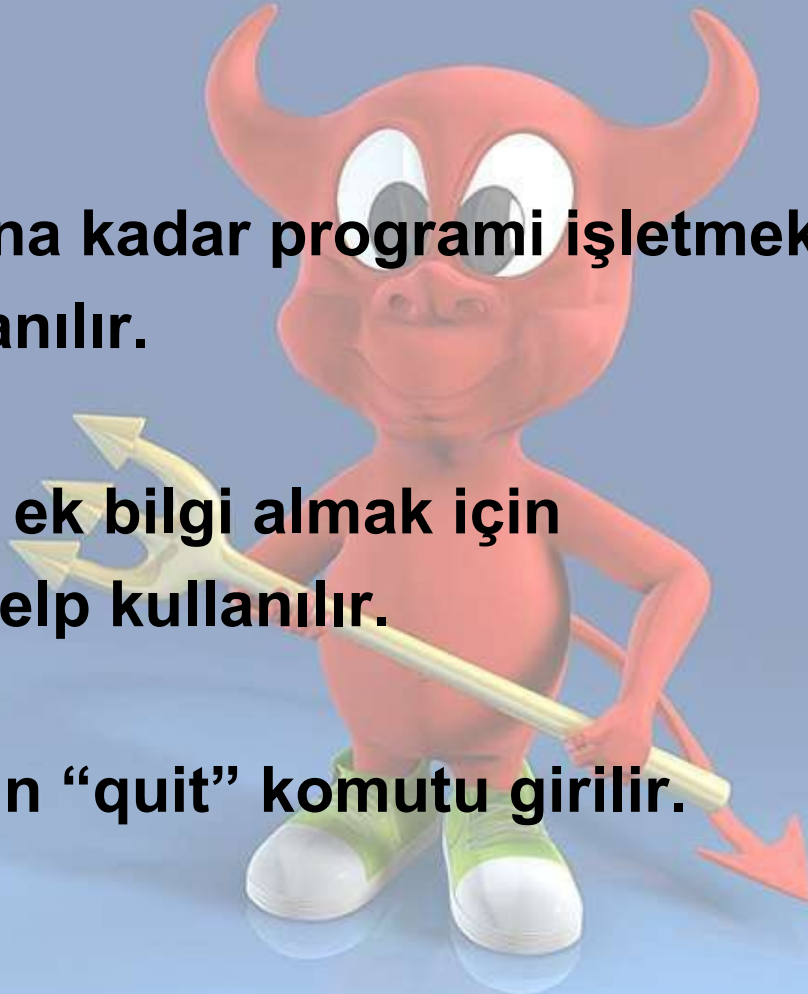
- Programın çalışmasına devam etmesini sağlamak için “continue” kullanılır. Kısaltma olarak ‘c’ kullanılması yeterlidir.
- Breakpoint’i silmek için delete kullanılır. N ifadesi kaçınıcı breakpointi silmek istediđinizi belirtir.

delete N



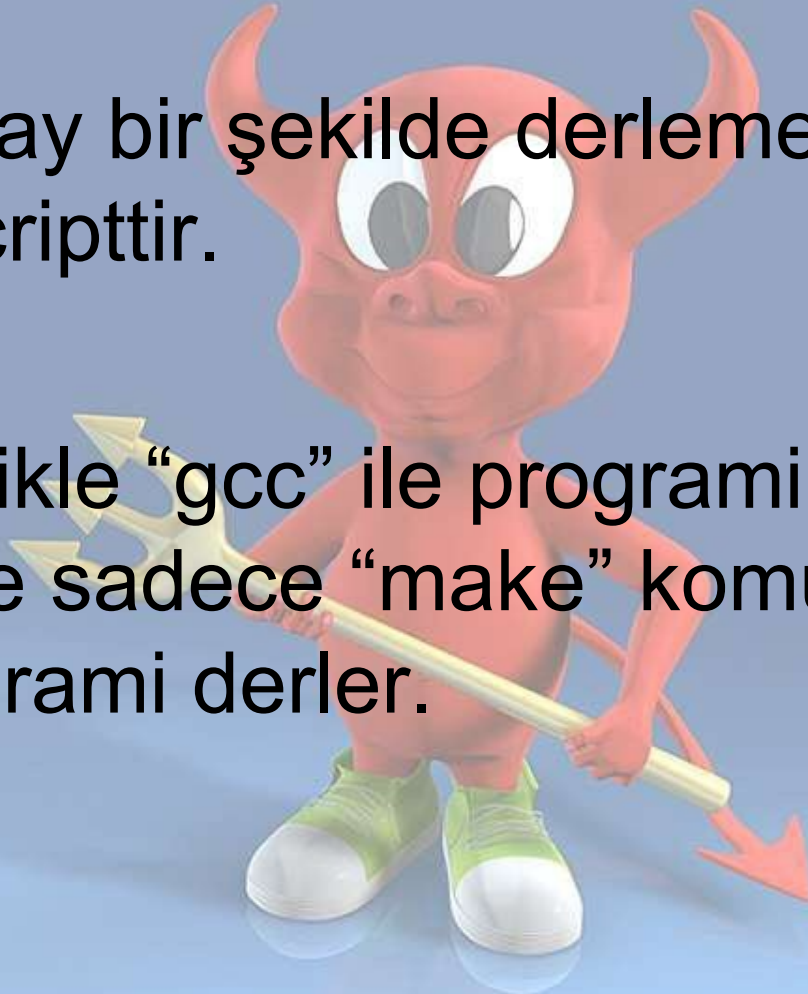
GDB ile Hata Ayıklama

- Fonksiyonun sonuna kadar programi işletmek için “finish ” komutu kullanılır.
- Komutlar hakkında ek bilgi almak için “info [komut]” veya help kullanılır.
- Gdb'den çıkmak için “quit” komutu girilir.



Makefile

- Programlari kolay bir şekilde derlemek için kullanılan bir scripttir.
- Kullanıcı böylelikle “gcc” ile programi derlemek yerine sadece “make” komutunu kullanarak programi derler.



Macro Kullanmadan Makefile

```
CC=gcc
```

```
CFLAGS=-c -Wall
```

```
all: hello
```

```
hello: main.o function.o hello.o
```

```
$(CC) main.o function.o hello.o -o hello
```

```
main.o: main.cpp
```

```
$(CC) $(CFLAGS) main.cpp
```

```
function.o: function.c
```

```
$(CC) $(CFLAGS) function.c
```

```
hello.o: hello.c
```

```
$(CC) $(CFLAGS) hello.c
```

```
clean: rm -rf *.o hello
```



Macro Kullanarak Makefile

CC = gcc

CFLAGS = -g -I. -I/usr/local/include

LIBS = -lpcap -liconv

LDFLAGS = -L/usr/local/lib

OBJS = hash.o main.o nstrstr.o msn.o sniff.o

all: \$(OBJS)

\$(CC) -o scanhill \$(OBJS) \$(LIBS) \$LDFLAGS

clean:

rm *.o scanhill



Mehmet Şerif DALGIÇ

ve

Bilişim Güvenliği Kulübü 'ne teşekkürler...





Sorular



Cihan Kömeçoğlu

EnderUNIX Yazılım Geliştirme Takımı

cihan@enderunix.org

<http://www.enderunix.org/cihan/>