

Bilgisayar Programlama I

DERS 1: C++ YÜKLENMESİ VE GENEL BİLGİLER

DR. HÜSEYİN BAHTİYAR

Neden Programlama?

2

- ▶ Bilgisayarlar bizim için iş yapmak için üretilmişlerdir.
- ▶ Bizler ise istediğimizi bilgisayara yaptırabilmek için onun anlayacağı dilden konuşmalıyız.
- ▶ Kullanıcılar ise programcıların bilgisayarın anlayacağı dilde yazdığı bu programları indirip kendi işleri için kullanırlar.



Ne
yapacağım?

Ne
yapacağım?

Ne
yapacağım?

Ne
yapacağım?



Programcılar gereklilikleri öngörürler

- ▶ Örnek olarak iPhone u alalım, iPhone app'leri bir pazardır.
- ▶ Bu appler 3 milyardan fazla kez yüklendi.
- ▶ Kimi programcılar işlerini bırakıp tam zamanlı iPhone app Geliştiricisi oluyorlar böylece pazardan daha fazla pay almaya Çalışıyorlar.



Kullanıcı ve Programcı arasındaki farklar

- ▶ Kullanıcılar bilgisayarı iş yapma aracı olarak görürler mesela excel kullanıcıları için; tablolarda hesaplama aracı, Word için; yazı aracı, harita, oyun vs...
- ▶ Programcılar ise bilgisayarın anlayacağı dili ve en efektif yolu öğrenen kişilerdir.
- ▶ Böylece ellerindeki programlama dillerini kullanarak yeni programlar yaratırlar.
- ▶ Bu programlar kimi zaman bir çok kullanıcının kullandığı programlar, kimi zaman ise küçük yardımcı programcıklar olabilir.

Programcılık? Neden?

5

- ▶ **Bazı işleri halletmek için, aslında bizler hem kullanıcı hem de programcıyız**
 - ▶ Parça parça bölünmüş bir dosyayı tek parça haline getirmek.
 - ▶ Gereksiz verileri sistemden temizlemek.
- ▶ **Başkalarının da kullanması için programlar üretmek**
 - ▶ Bir programdaki performans sorununu düzeltmek.
 - ▶ Web sitesine yorum kısmı eklemek.
 - ▶

Kullanıcı



<https://pixabay.com>

6

**Bilgisayar
Donanım + Yazılım**



Programcı

<https://openclipart.org/>

Veri

Bilgi

....

Network

Program yaratıcısının gözünde bizler son kullanıcı için program yaratan programcılarız. Bu arada veri, bilgi, network akışlarında olabilecek sorunlar da bizler tarafından çözülmelidir. Bu konuda bilgisayar donanım ve yazılımı bize yardımcı olur.

Kod? Yazılım? Program?

7

- ▶ **Sıralı bir şekilde saklanmış talimatlardır.**
 - ▶ Bizim zekamızın ufak bir kısmının bilgisayara aktarılmış halidir.
 - ▶ Bir sorunu çözdüğümüz zaman onu programa yazar ve başka biri ile paylaşıyoruz. Böylece programı paylaştığımız kişiler benzer sorunu çözmek için zaman ve enerjiden tasarruf etmiş olurlar.
- ▶ **Küçük yaratıcılıklar**
 - ▶ Özellikle kullanıcı deneyimiyle ilgili iyi bir iş yaptığımızda.

İnsanlar için program

8

- Buna en güzel örnek: Dans



C++ için program

9

```
#include <iostream>
using namespace std;

int main(){
cout << " ben bir c++ programıyım " <<endl;

return 0;

}
```

Donanım Mimarisi

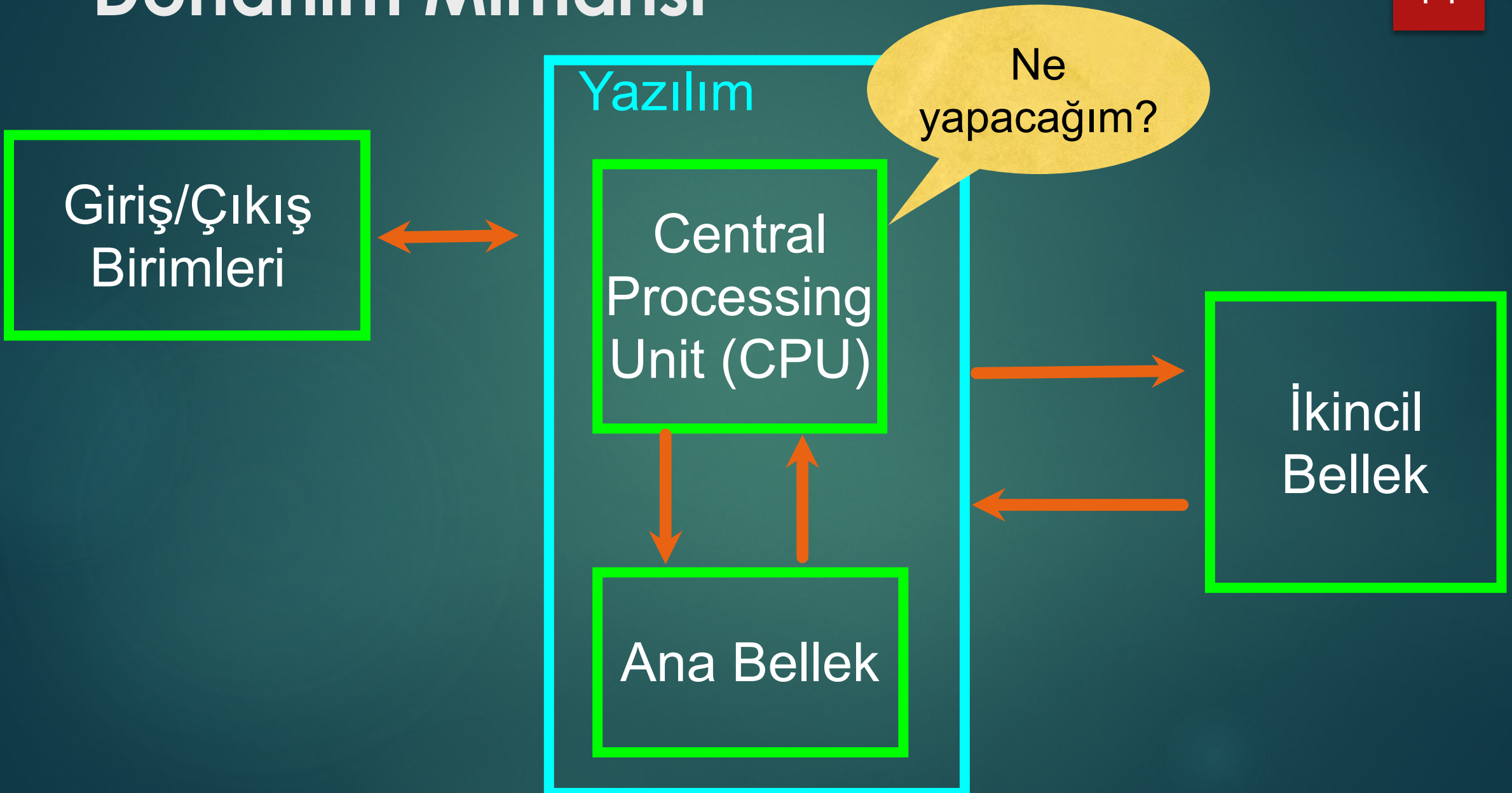
10



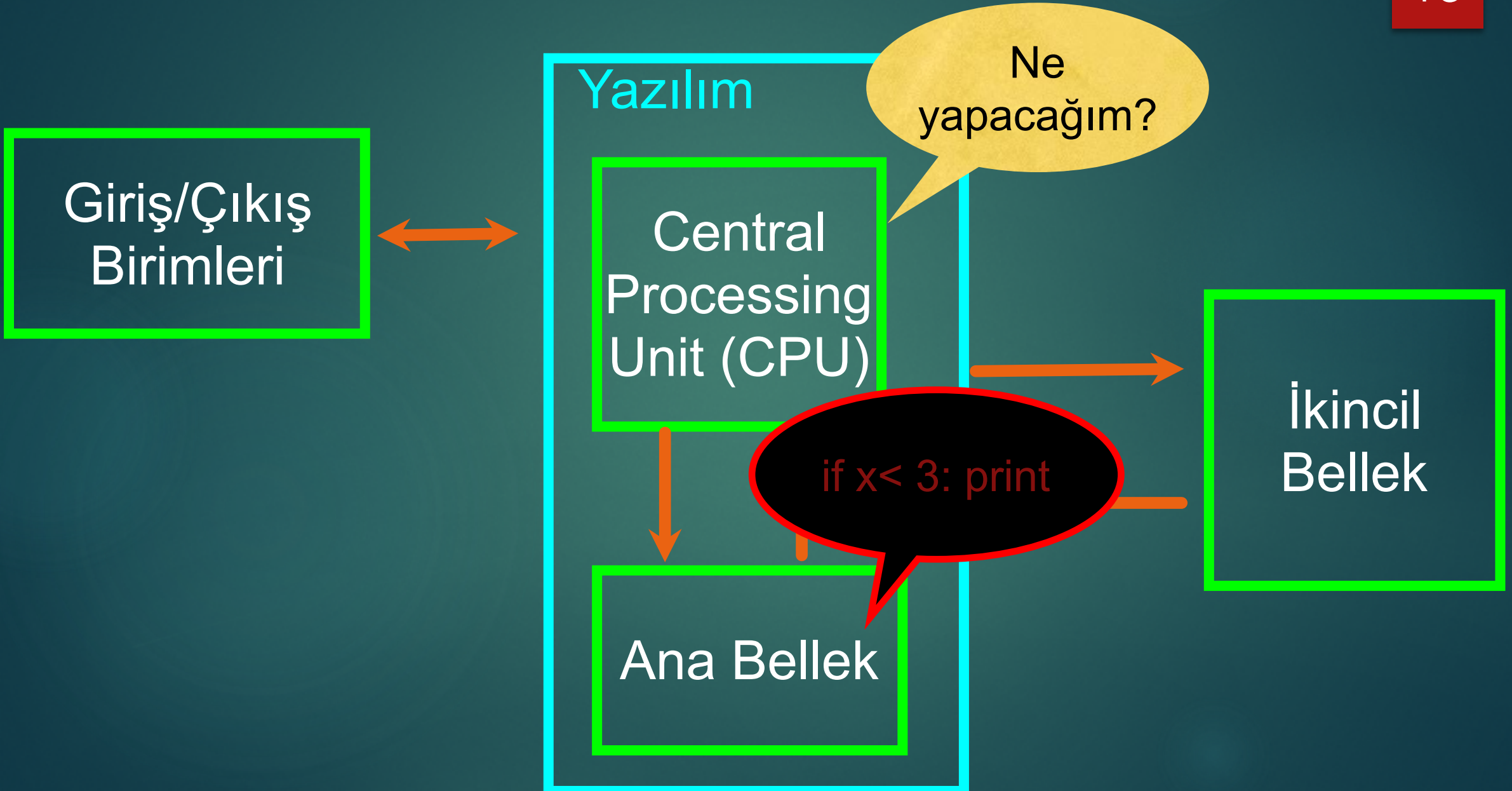
<http://upload.wikimedia.org/wikipedia/commons/3/3d/RaspberryPi.jpg>

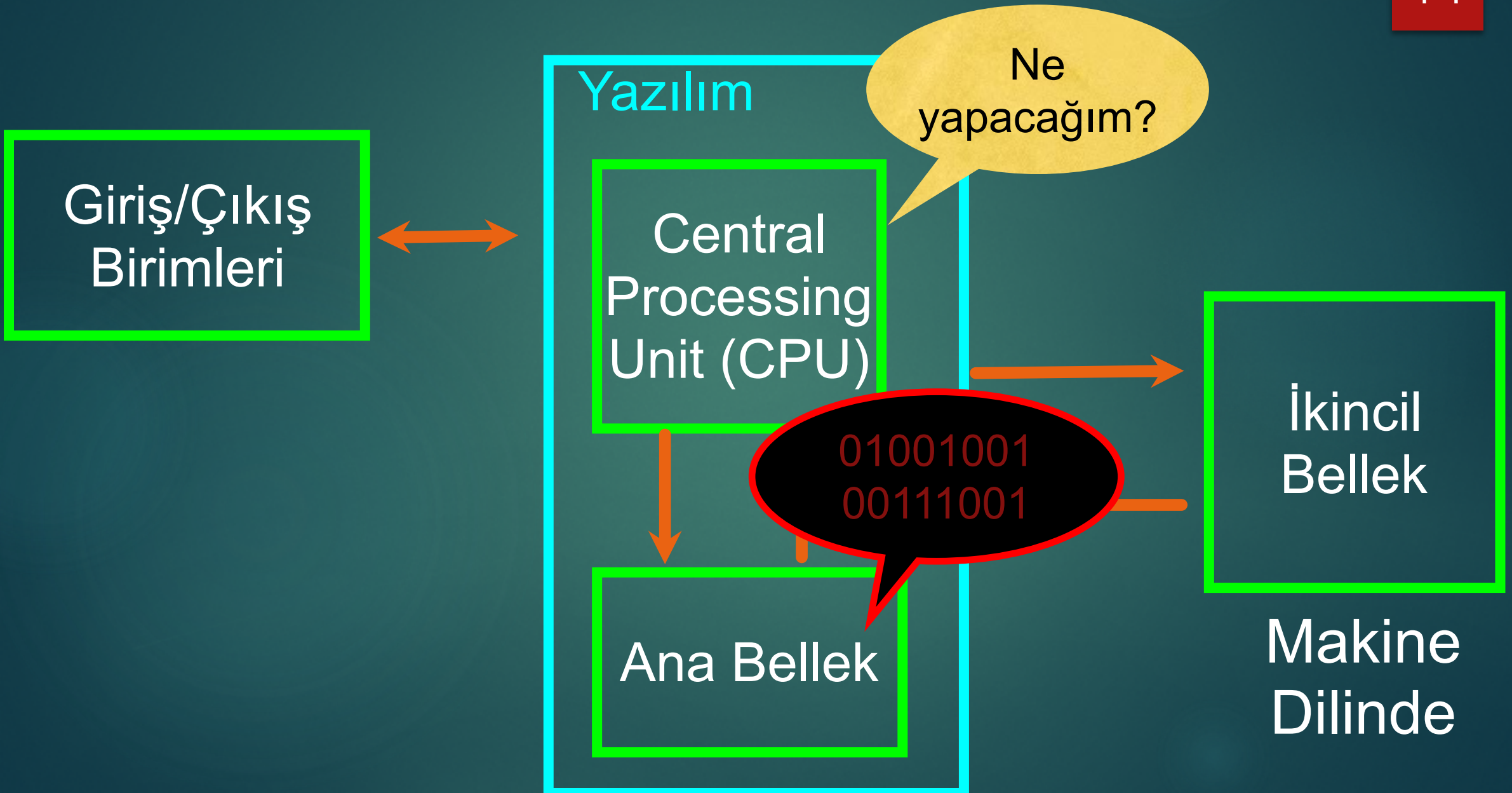
Donanım Mimarisi

11



- ▶ **CPU** : Programı çalıştıran kısımdır. CPU her zaman ne yapacağım? Diye görev bekleyen bir sistemdir. Aslında bilgisayarın beyni dememiz doğru olmaz, komut bekleyen bir yapıdır ancak çok çok hızlı çalışabilir.
- ▶ **Girdi** : Klavye, Mouse ...
- ▶ **Çıktı** : Ekran, hoparlör, printer vs...
- ▶ **Ana Bellek** : Hızlı küçük geçici saklama bölgesidir. Kapatıp açıldığında içindeki veriler kaybolur. RAM adı verilir.
- ▶ **İkincil Bellek** : Daha yavaş kalıcı saklama bölgesidir. Silinene kadar veriyi saklar. Sabit disk, usb bellek vs...





C++ Hakkında

15

- ▶ C++ 1980'lerin başında Bjarne Stroustrup tarafından geliştirilen C'yi kapsayan ve nesne yönelimli programlamaya olanak sağlayan sınıf tanımlarıyla ilerleten yaygın olarak kullanılan programlama dili. C++ genel amaçlı bir programlama dilidir.
- ▶ Arttırılmış C demektir ve C dilinin geliştirilmiş halini yansıtır.
- ▶ Her program bileşeninin bir nesneden oluştuğunu göz önüne alan bir dildir.



Programlama dilleri Seviyeleri

16

- ▶ **Çok yüksek seviyeli diller** ya da **görsel diller**: Access, Foxpro, Paradox, Xbase, Visual Basic, Oracle Forms
- ▶ **Yüksek seviyeli diller** veya **algoritmik diller**: Fortran, Pascal, Basic, Cobol
- ▶ **Orta seviyeli diller**: C, C++ (C Plus) , C# (C Sharp) Orta seviyeli diller daha az kayıpla makine diline çevrilebildiğinden daha hızlı çalışır.
- ▶ **Alçak seviyeli programlama dilleri**: Sembolik makine dili (Assembler).
- ▶ **Makine dili**: En aşağı seviyeli programlama dilidir (Saf makine dili tamamen 1 ve 0 lardan oluşuyor.)

Nasıl Yükleyebilirim?

17

- ▶ C++ kullanabilmek için programlama dilinin sisteme kurulu olması gerekir.
 - ▶ **Linux** : C++ zaten kurulu olarak gelmektedir. Pardus, ubuntu gibi işletim sistemlerinde C++ yüklemeye gerek yoktur.
 - ▶ **Windows** : <http://www.bloodshed.net/devcpp.html>
Adresinden Windows için olan versiyonunu indirebilirsiniz.

Laboratuvardaki bilgisayarlarda c++ yüklüdür. Şimdi konsol açıp çalışıp çalışmadığını deneyelim.

www.bloodshed.net/devcpp.html

Electronic Library LQCD ÇALIŞMA QFT Tübitak Projesi Günlük Kontrol Academic Jobs VW BETTLE-KIR Portakallı kurabi Other bookmarks

BloodshedSoftware

Providing Free Software to the internet community

Site Map
Home
Download
Email us
Get Bloodshed CD

Programmers
Free compilers list
Delphi
C and C++
Linux/Unix

Resources
Free URL submission
Add a resource

Dev-C++ 5 (currently beta)

 [Screenshot]

Bloodshed Dev-C++ is a full-featured Integrated Development Environment (IDE) for the C/C++ programming language. It uses Mingw port of GCC (GNU Compiler Collection) as it's compiler. Dev-C++ can also be used in combination with Cygwin or any other GCC based compiler.

Features are :

- Support GCC-based compilers
- Integrated debugging (using GDB)
- Project Manager
- Customizable syntax highlighting editor
- Class Browser
- Code Completion
- Function listing
- Profiling support
- Quickly create Windows, console, static libraries and DLLs
- Support of templates for creating your own project types
- Makefile creation
- Edit and compile Resource files
- Tool Manager
- Print support
- Find and replace facilities
- CVS support

Source code : Delphi 6 Source code of Dev-C++ is available for free under the GNU General Public License (GPL)

Authors : Colin Laplace, Mike Berg, Hongli Lai : Development
Mingw compiler: Mumit Khan, Jan Jaap van der Heidjen, Colin Hendrix and GNU coders.

System : Windows 95/98/NT/2000/XP

Status : Free Software (under the GNU General Public License)

Size : 13.5 Mb

Downloads : [Go to Download Page](#)

[Dev-C++ resources page \(libraries, sources, updates...\)](#)

Get Bloodshed CD 

Get it on Bloodshed CD

You can get all the software available on this site on a CD. This is the list of all software included on ONE CD:

- Dev-C++ 5 beta
- Dev-C++ 4 CD version
- Dev-C++ 4.01 update
- Dev-C++ for Linux
- QuickInstall 2.0
- Avi Creator 1.0
- Dev-C++ 4 sources
- Dev-Pascal 1.9 sources
- Multibox 4.0
- Fast Cleaner 1.0
- LaserWar
- LaserWar sources for Delphi
- Calc-By-Step 1.0
- Dev-C++ Packages
- + Goodies !

```

huseyin@huseyin-flzk:~$ which g++
/usr/bin/g++
huseyin@huseyin-flzk:~$ g++ --help
Usage: g++ [options] file...
Options:
  -pass-exit-codes      Exit with highest error code from a phase
  --help                Display this information
  --target-help          Display target specific command line options
  --help={common|optimizers|params|target|warnings|^[^}{joined|separate|undocumented}}[,...]
                        Display specific types of command line options
  (Use '-v --help' to display command line options of sub-processes)
  --version             Display compiler version information
  -dumpspecs             Display all of the built in spec strings
  -dumpversion           Display the version of the compiler
  -dumpmachine           Display the compiler's target processor
  -print-search-dirs     Display the directories in the compiler's search path
  -print-libgcc-file-name Display the name of the compiler's companion library
  -print-file-name=<lib> Display the full path to library <lib>
  -print-prog-name=<prog> Display the full path to compiler component <prog>
  -print-multiarch       Display the target's normalized GNU triplet, used as
                        a component in the library path
  -print-multi-directory Display the root directory for versions of libgcc
  -print-multi-lib        Display the mapping between command line options and
                        multiple library search directories
  -print-multi-os-directory Display the relative path to OS libraries
  -print-sysroot          Display the target libraries directory
  -print-sysroot-headers-suffix Display the sysroot suffix used to find headers
  -Wa,<options>           Pass comma-separated <options> on to the assembler
  -Wp,<options>           Pass comma-separated <options> on to the preprocessor
  -Wl,<options>           Pass comma-separated <options> on to the linker
  -Xassembler <arg>      Pass <arg> on to the assembler
  -Xpreprocessor <arg>   Pass <arg> on to the preprocessor
  -Xlinker <arg>          Pass <arg> on to the linker
  -save-temps            Do not delete intermediate files
  -save-temps=<arg>      Do not delete intermediate files
  -no-canonical-prefixes Do not canonicalize paths when building relative
                        prefixes to other gcc components
  -pipe                  Use pipes rather than intermediate files
  -time                  Time the execution of each subprocess
  -specs=<file>           Override built-in specs with the contents of <file>
  -std=<standard>         Assume that the input sources are for <standard>
  --sysroot=<directory>  Use <directory> as the root directory for headers
                        and libraries

```

Neden C++?

20

► NEDEN C++ ?

- OOP (Object Oriented Programming- Nesneye yönelik programlama)
 - Portatiflik
 - Kısalık ve özlük
 - Modülerlik
 - Uyumluluk
 - Hız
- Fizikte birçok araştırma alanında C++ tabanlı programlar kullanılmaktadır
- GEANT4 : <http://geant4.web.cern.ch/geant4/>
 - ROOT: <http://root.cern.ch/>

İlk Program

21

- Öncelikle bir text editör açalım daha sonra ilk programımızı yazalım.

```
#include <iostream>

//ilk programım

using namespace std;

int main(){
cout << " ben bir c++ programıyım " <<endl;
return 0;
}
```

- Dosya adını Ders01-01.cpp olarak kayıt edelim.

İlk Program

22

- Öncelikle bir text editör açalım daha sonra ilk programımızı yazalım.

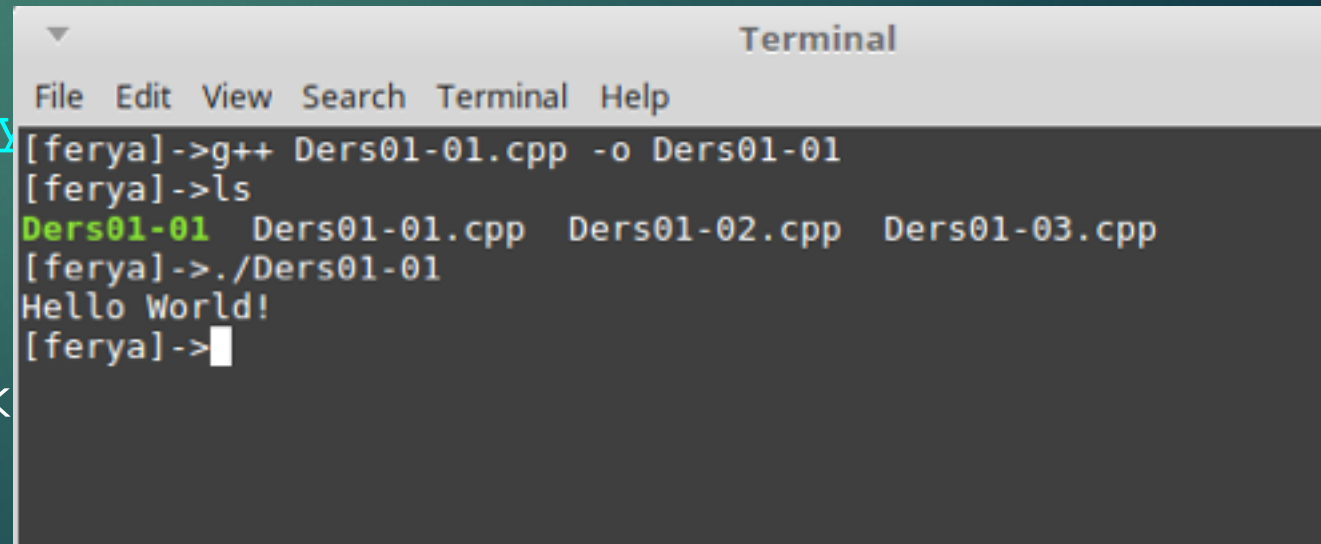
```
#include <iostream>

//ilk programım

using namespace std;

int main(){
cout << " ben bir c++ programıyım" << endl;
return 0;
}
```

- Dosya adını Ders01-01.cpp olarak k



```
Terminal
File Edit View Search Terminal Help
[ferya]->g++ Ders01-01.cpp -o Ders01-01
[ferya]->ls
Ders01-01 Ders01-01.cpp Ders01-02.cpp Ders01-03.cpp
[ferya]->./Ders01-01
Hello World!
[ferya]->
```

İlk Program

23

```
#include <iostream>
```

Ön işlemci için yönlendirme (programa ben bu komutları kullanacağım diyoruz)

```
//ilk programım
```

Yorum satırı // işaretinin sağındaki yazı derleyici tarafından önemsenmez atlanır. Bu kısım programı okuyanlar için

```
using namespace std;
```

C++ kütüphanesinin standart elemanlarını kullanacağım

```
int main(){
```

Ana programa başlıyorum

```
cout << " ben bir c++ programıyım " <<endl;
```

```
return 0;
```

```
}
```

Ana programı kapattım

İlk Program

24

Alt satıra geçme komutu.

```
cout << " ben bir c++ programıyım " <<endl;
```

Ekrana yazdırma komutu

Ekranda göreceğimiz yazı

Bu işlem ; ile sonlandırılır.

```
return 0;
```

Ana fonksiyon ile işim bitti

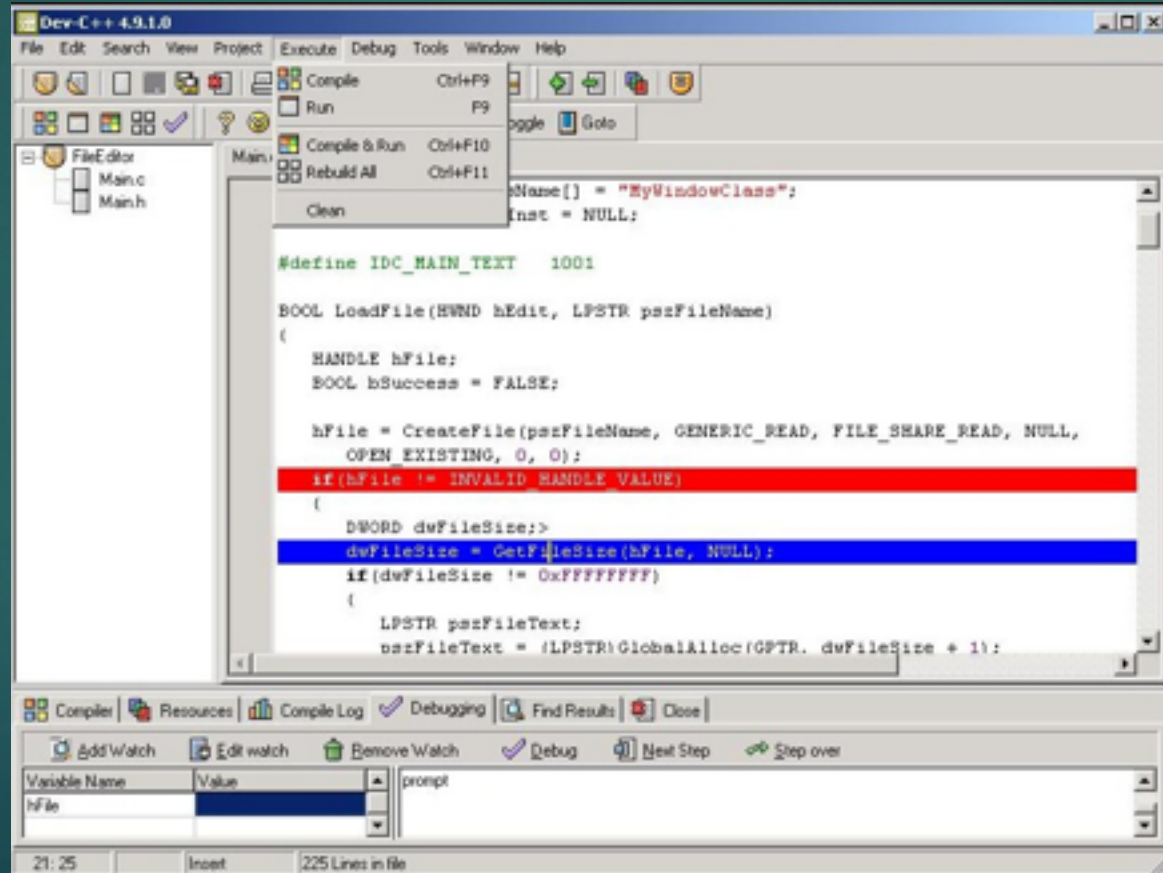
Programı Derlemek

25

Linux:

`g++ dosyaadi.cpp -o çalıştırılacak_program_adı`

Windows:



Kaynaklar

26

- ▶ <http://www.cplusplus.com/doc/tutorial/>
- ▶ www.py4e.com
- ▶ C++ How to program (Deitel)
- ▶ Ferhat Özok Bilgisayar Programlama I ders notları