مبانی برنامه نویسی در محیط C

مراجع درس

- مونس علی فرمنش، "برنامه نویسی جامع به زبان C"، انتشارات جهاد دانشگاهی دانشگاه صنعتی امیر کبیر
- عین اله جعفر نژاد، "برنامه نویسی به زبان C"، انتشارات جهاد دانشگاهی مشهد

مطالب امروز...

- محیط برنامه نویسی متمرکز IDE
- معرفی و آشنایی با زبان برنامه نویسی C
- متغیرهای عددی

محیط برنامه نویسی متمرکز IDE

- روش متداولی که برای تولید و توسعه برنامه ها مورد استفاده قرار میگیرد استفاده از محیط برنامه نویسی متمرکز (IDE:Integrated Design Environment) میباشد
- در این سیستم تمام عملگرهای ضروری برای ایجاد یک برنامه در یک صفحه نمایش واحد قرار دارند. عملگرها از منو انتخاب میشوند.

File Edit Search Run Compile Debug Project Options Window Help

NONAME00.CPP

HelloWorld - Microsoft Visual C++ - [HelloWorld.cpp]

File Edit View Insert Project Build Tools Window Help

(Globals) | (All global members) | main

HelloWorld classes
Globals
main(int argc, char

ClassView | FileView

```
// HelloWorld.cpp : Defines the entry point for the console appli
//

#include "stdafx.h"

int main(int argc, char* argv[])
{
    printf("Hello World!\n");
    return 0;
}
```

Build | Debug | Find in Files 1 | Find in Files 2

Ready | Ln 6, Col 1 | REC COL OVR READ

F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

زبانهای دارای کامپایلر

- برنامه ورودی فقط یکبار به زبان ماشین تبدیل شده و به شکل یک فایل اجرایی در می‌آید
- فایل زبان ماشین به سرعت اجرا میشود
- بخشی از IDE، برنامه‌ای است با عنوان کامپایلر که کد برنامه قابل درک توسط انسان را به زبان ماشین تبدیل میکند
- بخش دیگر IDE عملیات الحاق (link) را انجام میدهد.

فرایند الحاق

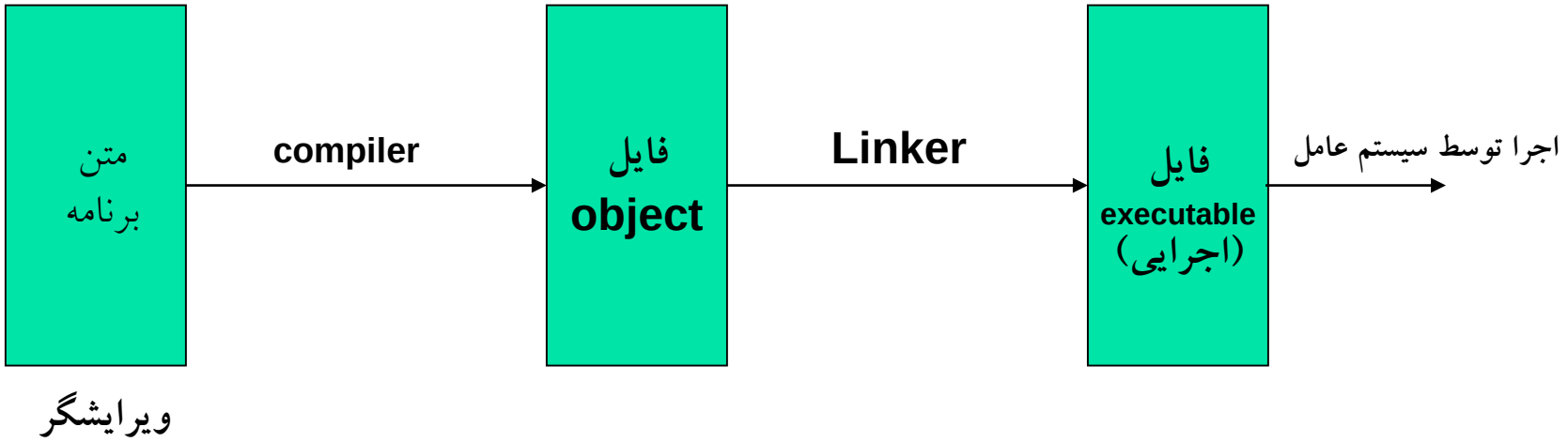
- فایل تولید شده توسط کامپایلر، قابل اجرا نمی‌باشد.
  - ممکن است برنامه نوشته شده "فایل کتابخانه‌ای" داشته باشد
  - ممکن است برنامه از فایلهای مجزا و متعددی تشکیل شده باشد که همزمان کامپایل نشده‌اند.

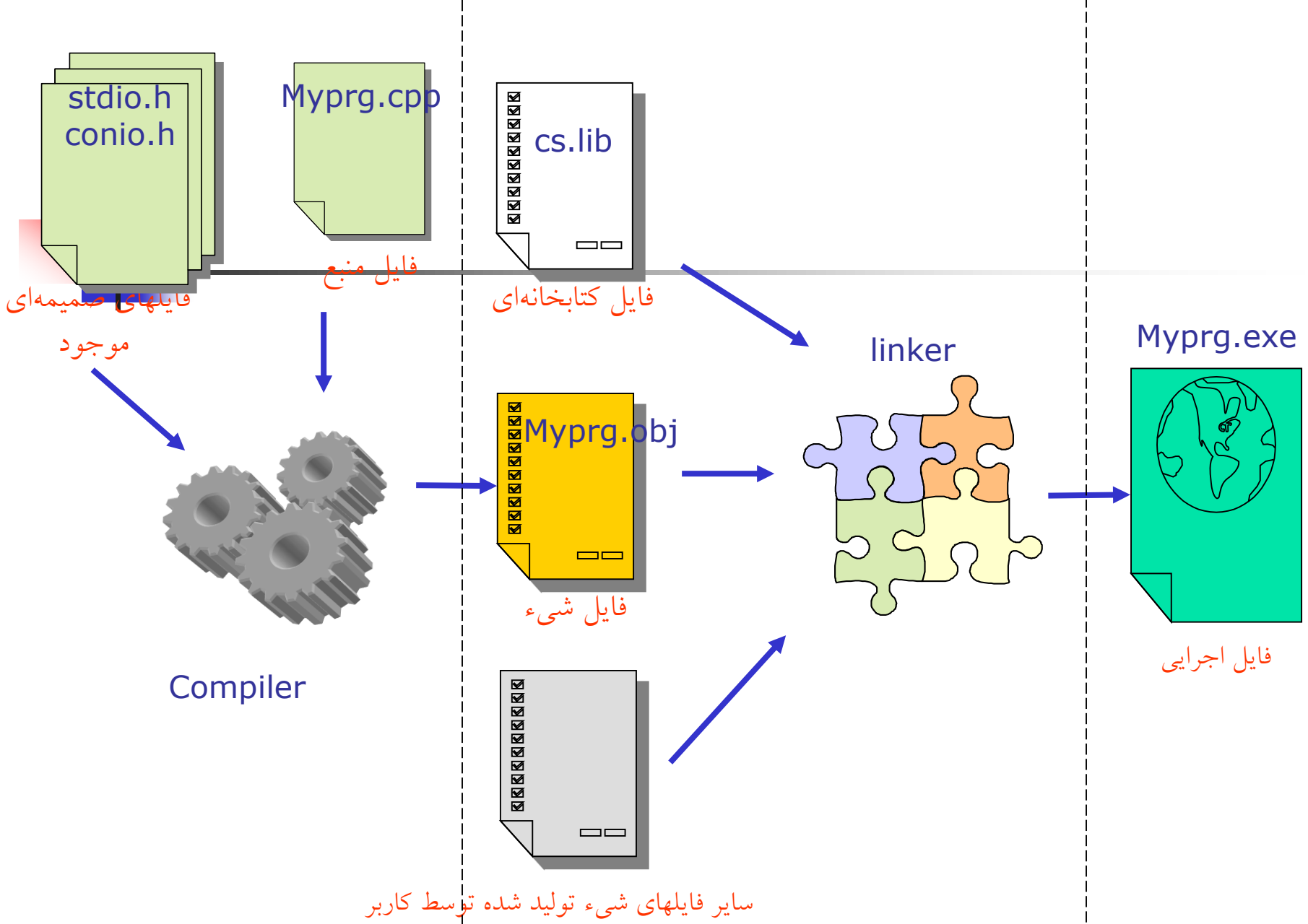
فرایند ساخت (Build)

- قدم۱) فایل منبع (.cpp) در محیط IDE ایجاد میشود.
- قدم ۲) این فایل به کامپایلر رفته و تولید فایل (.obj) میکند.
- قدم ۳) حاصل کار به الحاق‌گر ارسال شده و فایلهای اجرایی (.exe) ایجاد میشوند

مراحل ایجاد یک برنامه

- **Build=compile+link**
- **Builder=compiler+linker**




stdio.h
conio.h
Myprg.cpp
cs.lib
فایل منبع
فایل کتابخانه‌ای
موجود
linker
Myprg.exe
Myprg.obj
فایل شیء
Compiler
فایل اجرایی
سایر فایلهای شیء تولید شده توسط کاربر

Hello World اولین برنامه:

```
1  // Hello World program
2  #include <stdio.h>
3  int main()
4  {
5     printf("Hello World\n");
6     return 0;
7  }
```

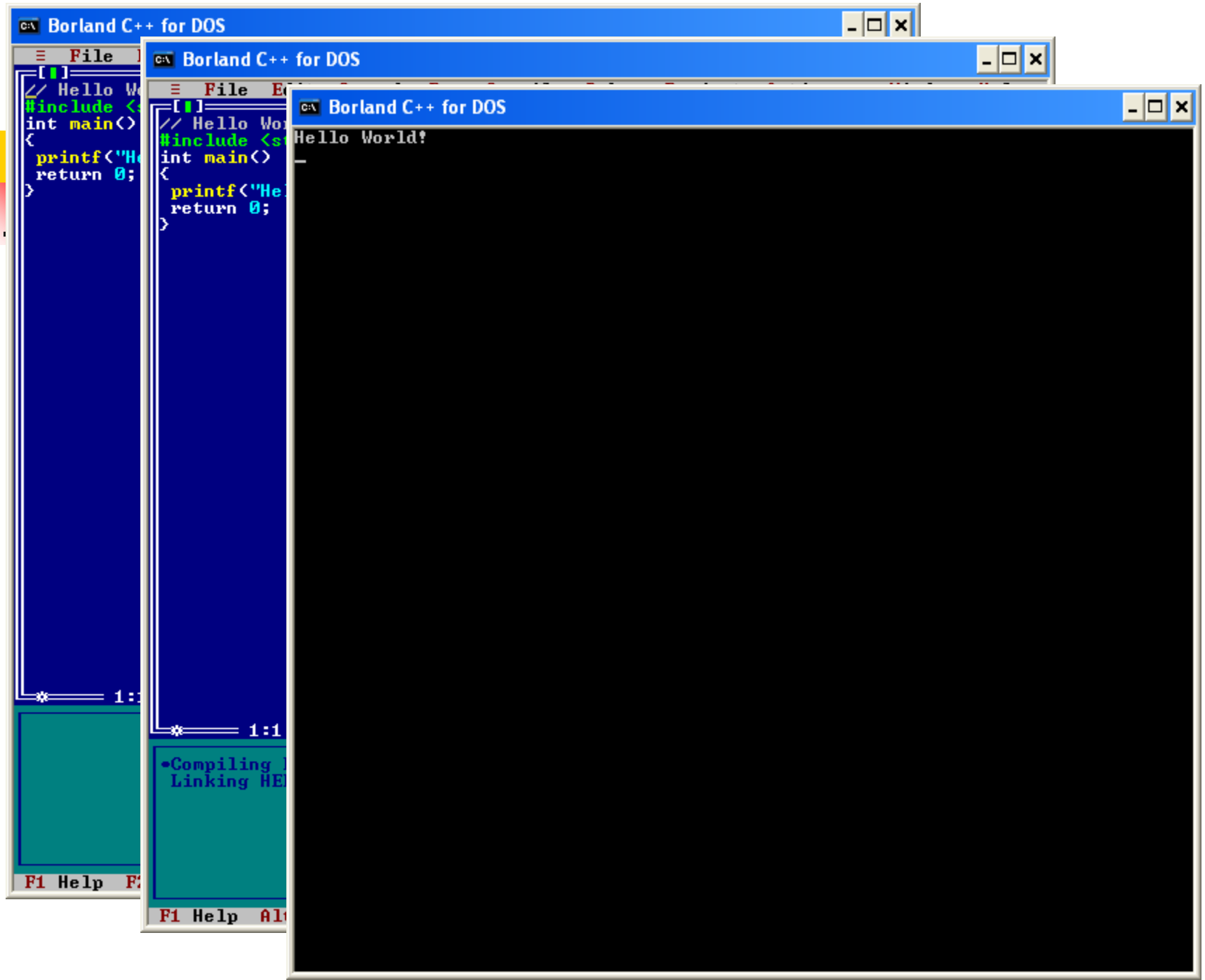
comment

Allows access to an I/O library

Starts definition of special function main()

output (print) a string

Program returns a status code (0 means OK)


Borland C++ for DOS
File
// Hello Wo
#include <s
int main()
{
printf("He
return 0;
}
1:1
F1 Help
Borland C++ for DOS
File
// Hello Wor
#include <st
int main()
{
printf("Hel
return 0;
}
1:1
•Compiling
Linking HE
F1 Help
Borland C++ for DOS
Hello World!

نکاتی در خصوص زبان برنامه نویسی C

- اکثر دستورات به ;(سمیکالن) ختم میشوند
- برنامه با تابع main آغاز میشود.
- کد مربوط به هر تابع بین دو آکولاد {} قرار میگیرد.
- استفاده از آکولاد برای ایجاد یک بلوک در این زبان استفاده میشود.
- حساس بودن به حالت حروف (case sensitivity)
  - در نام گذاری توابع و متغیرها حروف کوچک و بزرگ با هم متفاوت هستند ALI ≠ALi ≠Ali ≠ali

Comments

- توضیحات قسمتی از متن برنامه است که توسط کامپایلر ترجمه نمی شود و برای کمک به کسانی است که برنامه را می خوانند. (خوانا کردن برنامه)
- توضیحات تک خطی با // شروع می شوند.
  - در هر جای خط از برنامه با مشاهده این علامت مابقی خط نادیده گرفته میشود.
- توضیحات چند خطی با */ شروع و با /* پایان می یابند.

Comment Example

```
// program #1.1
// This program is for comment information
#include <stdio.h>
/* This program displays a real number
   with 27 decimal places.
*/
int main()
{
   printf("1.000000000000000000000001");
   return 0; // return the status code to OS

}
```

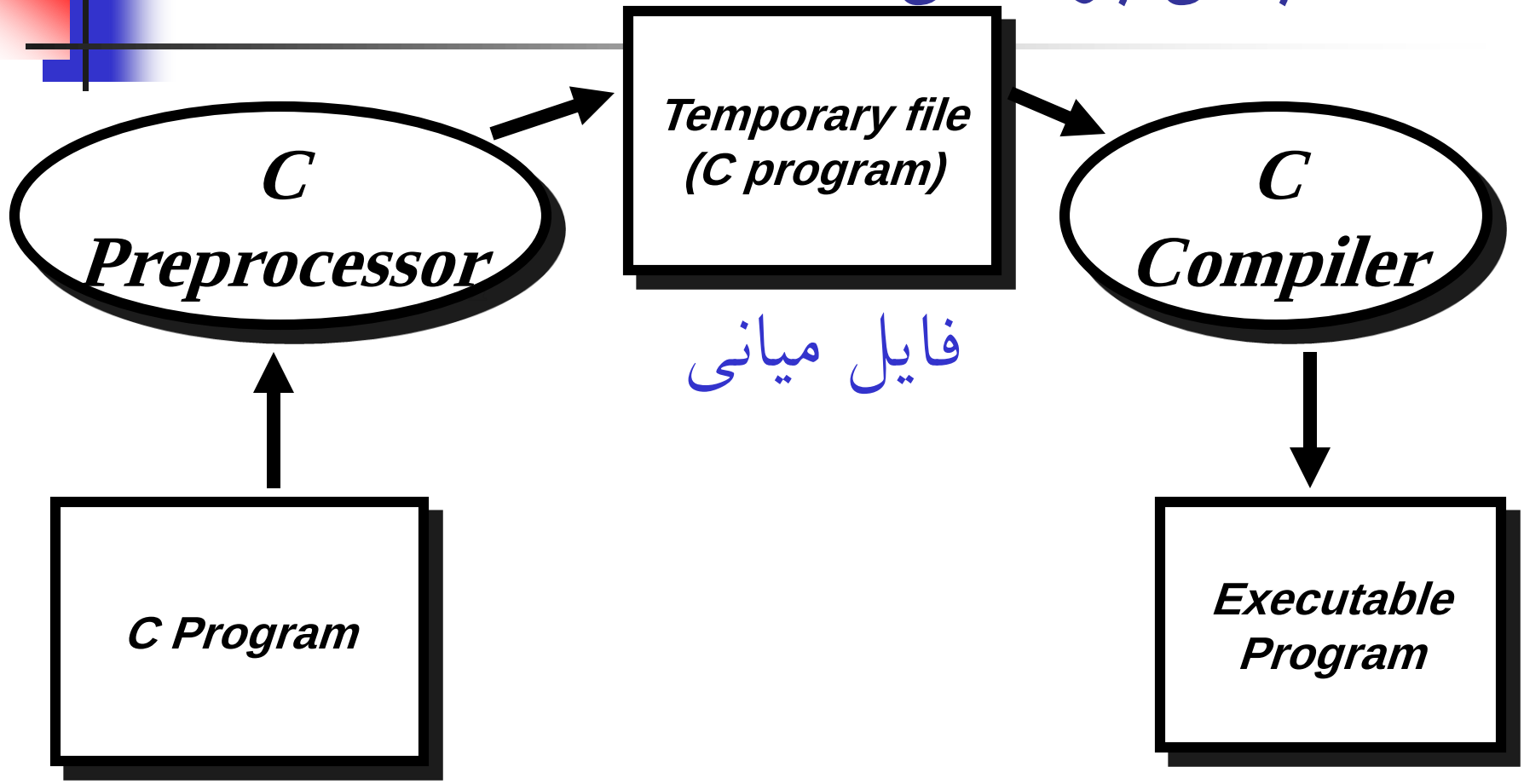
Includes

- اعلان #include <stdio.h> باعث اضافه شدن محتویات فایل stdio.h به فایل فعلی می گردد. این عمل قبل از کامپایل کد انجام میگردد.
- بدین ترتیب، فایلهای استاندارد کتابخانه ای C ، که حاوی تعاریف و توابع از پیش انجام شده مهمی هستند را می توان به برنامه اضافه کرد و استفاده نمود.
- همچنین، می توانید فایلهایی را که خودتان قبلاً نوشته اید به برنامه فعلی اضافه کنید:
- #include "myfile.h"

C Preprocessor

- کامپایلر C به طور اتوماتیک یک پیش پردازنده را صدا می‌کند که #include ها و راهنماها (directives) را پردازش می‌کند.
- برای اجرای پیش پردازنده لازم نیست که شما کار خاصی انجام دهید. این امر به طور اتوماتیک هنگام کامپایل برنامه انجام می‌گردد.

Preprocessing پیش پردازش



The Preprocessor

- تمام خطوطی که با # (number sign) شروع می شوند توسط پیش پردازنده پردازش می‌شوند.
- ممکن است پیش پردازنده این خطها را با چیز دیگری عوض کند.
  - مثلاً include با محتوای فایلی که الحاق شده است عوض میشود.
  - بقیه راهنما ها به پیش پردازنده می‌گویند که به دنبال یک الگوی خاص در برنامه بگردند و پردازشهایی مخصوصی روی آن انجام دهند. به این نوع راهنماها ماکرو میگویند.

#define Example
(macro)

```
#define square(a) (a * a)

y = square(x);
```

becomes `y = (x * x);`

```
z = square(y*x);
```

becomes `z = (y*x * y*x);`

Some common includes

- Basic I/O: stdio.h
- I/O : conio.h
- Standard Library: stdlib.h
- Time and Date support: time.h
- Mathematical library: math.h

تابع main()

- main() تابعی است که هنگام اجرای برنامه از طریق سیستم عامل فراخوانی میشود. و اولین تابعی است که اجرا میگردد.
- برنامه های C از تعدادی تابع تشکیل می شوند.
- هر برنامه نوشته شده به زبان C حتماً باید تابع main داشته باشد.

تابع printf()

- این تابع برای نمایش اطلاعات برروی صفحه نمایش بکار گرفته میشود.
- این تابع قادر است خروجی را با توجه به فرمت مورد نیاز تولید نماید.
- نحوه استفاده از این تابع به شکل زیر میباشد
- printf(<رشته کنترلی>[,<متغیرها>]);

این بخش میتواند در دستور نباشد (optional)

دستور return 0;

- توابع در C میتوانند یک مقدار خروجی داشته باشند
- خروجی هر تابع توسط دستور return به سیستم عامل اعلام میگردد
- از آنجا که طبق تعریف خروجی تابع main از نوع عدد صحیح (int) است، باید در انتهای اجرای این تابع یک مقدار صحیح به سیستم عامل بعنوان خروجی تابع معرفی شود در غیر اینصورت کامپایلر پیغام خطا میدهد.

تمرین

- با توجه به مطالب عنوان شده، برنامه ای بنویسید که خروجی به شکل زیر داشته باشد

Hello world!

I am a c programmer

حل

```
// program #2 solution 2
#include <stdio.h>
int main()
{
    printf("Hello world! \nI am a c programmer");
    return 0;
}
```

وجود \n در رشته کنترلی باعث
انتقال مکان نما به سطر بعد میگردد

انواع خطا

- **error**: به خطاهای برنامه نویسی **error** می گویند.
- انواع خطاها در برنامه نویسی:
  - خطاهای زمان **compile**(**compile errors**):
    - مانع کامپایل صحیح برنامه می شوند.
  - خطاهای زمان **link** (**Link errors**):
    - برای کامپایل مزاحمتی ایجاد نمی کنند اما مانع **Link** برنامه می شوند.
  - خطاهای زمان اجرا: (**Run time errors**):
    - کامپایل و **Link** با موفقیت انجام می شود ولی اجرای برنامه دچار اشکال می شود.

error

- حسن سیب را خورد.
- هسن سیب را خورد.
  - متناظر با خطای کامپایل
- را حسن خورد سیب.
  - متناظر با خطای Link
- سیب حسن را خورد.
  - متناظر با خطای زمان اجرا
    - مانند تقسیم بر صفر