

# Programmazione e Laboratorio di Programmazione

## Integrazione I

### Il compilatore e le sue opzioni

# Generazione dell'eseguibile

## 1. Preprocessing

1.1 rimozione dei commenti

1.2 interpretazione delle direttive per il preprocessore

## 2. Compilazione

2.1 generazione del codice assembler

2.2 generazione dei moduli oggetto

## 3. Linking

3.1 combinazione dei diversi moduli oggetto in un unico eseguibile

# Il preprocessing

```
1 /*  
2 ** Questo è un semplice programma di prova  
3 */  
4  
5 #define IVA 0.20  
6  
7 int main ()  
8 {  
9   int x;  
10  
11   x = IVA * 20000;  
12  
13   return(0);  
14 }
```

**interpretazione delle direttive per il preprocessore**

**rimozione dei commenti**

```
1 int main ()  
2 {  
3   int x;  
4  
5   x = 0.20 * 20000;  
6  
7   return(0);  
8 }
```

# Compilazione

- **Generazione del codice assembler:**  
traduzione del codice sorgente ricevuto dal preprocessore in linguaggio assembler

```
1 #include <stdio.h>
2
3 int main ()
4 {
5     printf("Hello World!");
6
7     return(0);
8 }
```

```
IDEAL
MODEL SMALL
STACK 100h
DATASEG
    HW      DB      "hello, world", 13, 10, '$'
CODESEG
Begin:
    MOV AX, @data
    MOV DS, AX
    MOV DX, OFFSET HW
    MOV AH, 09H
    INT 21H
    MOV AX, 4C00H
    INT 21H
END Begin
```

# Compilazione

- **Linguaggio assembler:**

linguaggio di basso livello che fornisce un minimo livello di astrazione (es., utilizzo di codici mnemonici in luogo di codici numerici) rispetto alle istruzioni eseguibili dal microprocessore

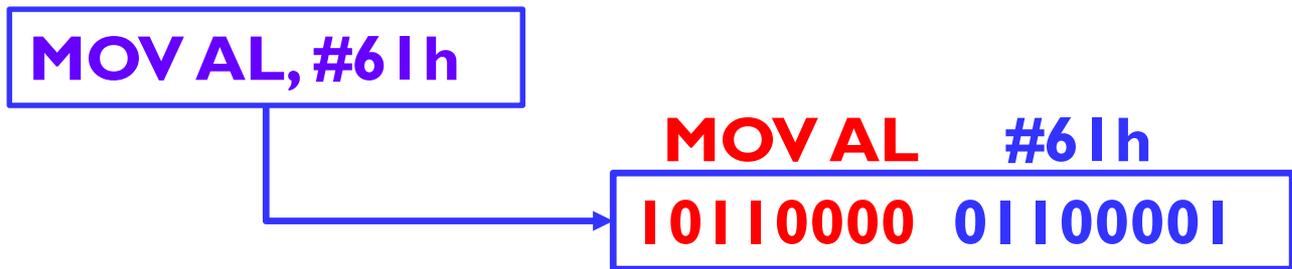
- **Esempio:**

```
MOV AL, #61h
```

memorizza il valore  
esadecimale 61 nel registro di  
nome AL

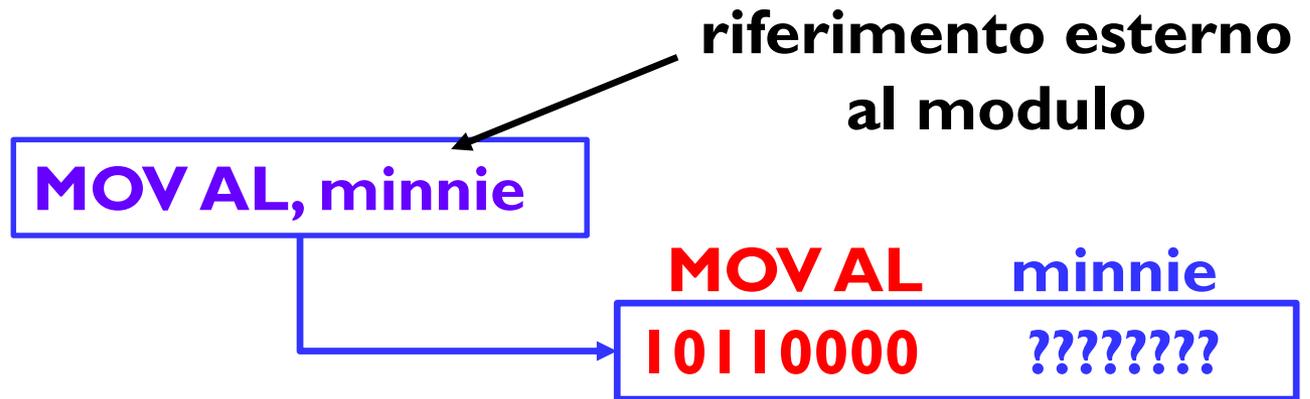
# Compilazione

- **Generazione dei moduli oggetto :**  
traduzione del codice assembler in una sequenza di istruzioni macchina non ancora eseguibili a causa, tra l'altro, di riferimenti esterni
- **Esempio:**

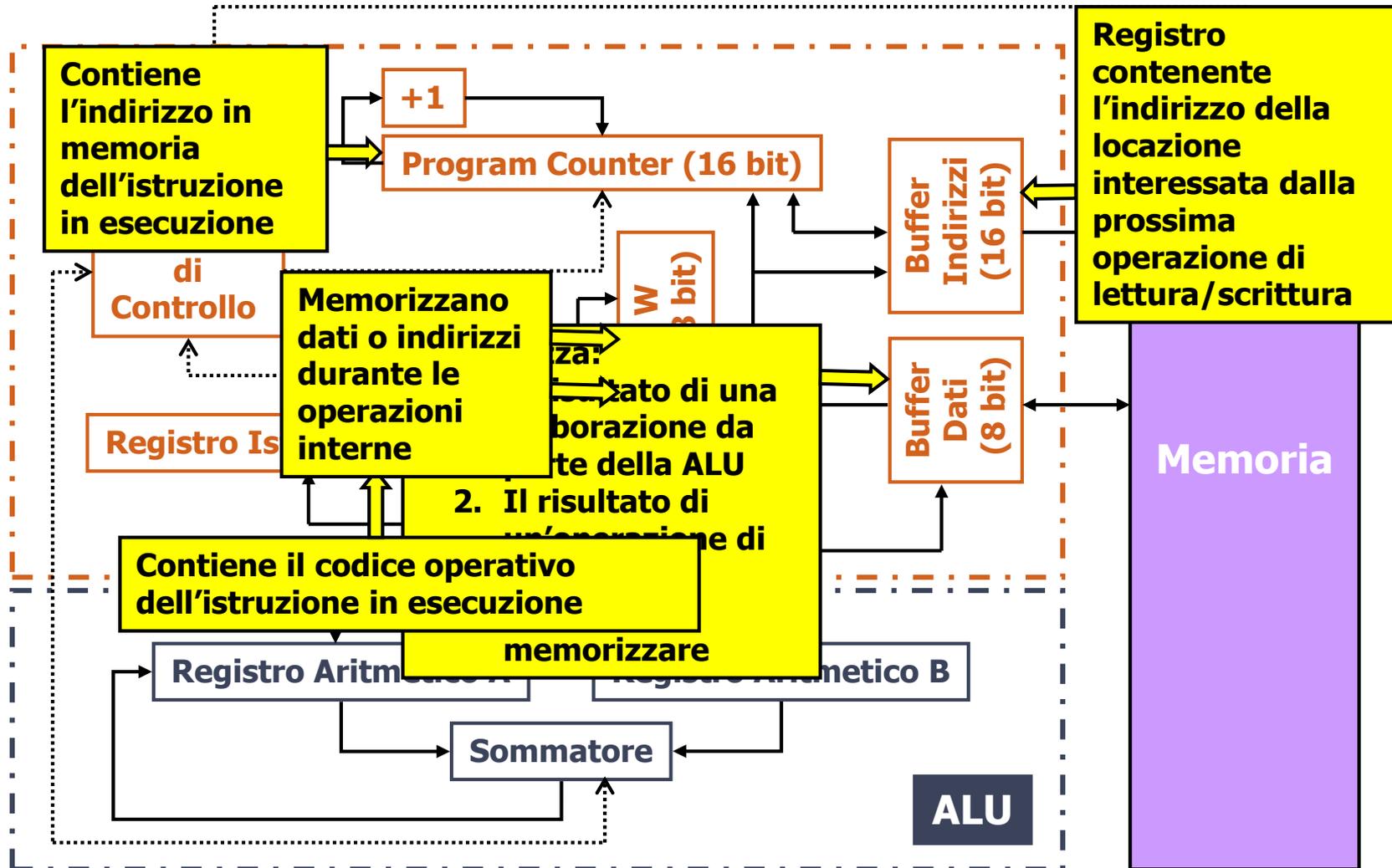


# Compilazione

- **Riferimenti esterni:**



# CPU: l'architettura



# Linking

- **Problema:**

esistenza all'interno dei moduli oggetto di riferimenti ad altri moduli (chiamate di funzioni, salti, variabili, ...)

- **Linker:**

programma che combina uno o più moduli oggetto generati durante la fase di compilazione in un unico file eseguibile

# Riconoscere i file

Estensione	Descrizione
<b>.c</b>	<b>file sorgente</b>
<b>.h</b>	<b>file generalmente “copiato” dal preprocessore all’interno di un file sorgente</b>
<b>.o</b>	<b>modulo oggetto</b>

# Opzioni del compilatore

- **Determinare la versione del compilatore :**

**gcc -v**

```
C:\windows\system32>gcc -v_
```

```
Reading specs from C:/Programmi/Mingw/bin/./lib/gcc/mingw32/3.4.5/specs
Configured with: ./gcc-3.4.5-20060117-3/configure --with-gcc
--with-gnu-ld --with-gnu-as --host=mingw32 --target=mingw32 --prefix=/mingw --enable-threads --disable-nls --enable-languages=c,c++,f77,ada,objc,java --disable-win32-registry --disable-shared --enable-sjlj-exceptions --enable-libgcj --disable-java-awt --without-x --enable-java-gc= Boehm --disable-libgcj-debug --enable-interpreter --enable-hash-synchronization --enable-libstdc++-debug
Thread model: win32
gcc version 3.4.5 (mingw-vista special r3)
```

# Opzioni del compilatore

- **Assegnare un nome al file eseguibile:**

**gcc nome\_sorgente -o nome\_eseguibile**

```
Directory di D:\Codice\Lezioni\Lezione_VII
27/11/2009  20.55    <DIR>      .
27/11/2009  20.55    <DIR>      ..
12/11/2009  19.08                170 compilabile.c
                1 File                170 byte
                2 Directory  199.721.558.016 byte disponibili
```

```
D:\Codice\Lezioni\Lezione_VII>gcc compilabile.c -o eseguibile
```

```
Directory di D:\Codice\Lezioni\Lezione_VII
27/11/2009  20.59    <DIR>      .
27/11/2009  20.59    <DIR>      ..
12/11/2009  19.08                170 compilabile.c
27/11/2009  20.59                15.883 eseguibile.exe
                2 File                16.053 byte
                2 Directory  199.721.541.632 byte disponibili
```

# Opzioni del compilatore

- **Generazione modulo oggetto:**

**gcc -c nome\_sorgente**

```
Directory di D:\Codice\Lezioni\Lezione_VII
27/11/2009  20.55    <DIR>      .
27/11/2009  20.55    <DIR>      ..
12/11/2009  19.08                170 compilabile.c
                1 File                170 byte
                2 Directory  199.721.558.016 byte disponibili
```

```
D:\Codice\Lezioni\Lezione_VII>gcc -c compilabile.c
```

```
Directory di D:\Codice\Lezioni\Lezione_VII
27/11/2009  21.06    <DIR>      .
27/11/2009  21.06    <DIR>      ..
27/11/2009  21.05                212 compilabile.c
27/11/2009  21.06                574 compilabile.o
                2 File                786 byte
                2 Directory  199.721.533.440 byte disponibili
```

# Opzioni del compilatore

- **Chiamare il preprocessore**

**gcc -E nome\_sorgente -o nome\_destinazione**

```
D:\Codice\Lezioni\Lezione_VII>type compilabile.c
```

```
// sorgente: Lezione_VII\compilabile.c
// file compilabile senza errori e senza warning
#define STRINGA "Hello world!"
#include <stdio.h>
int main ()
{
    printf(STRINGA);
    return(0);
}
```

```
D:\Codice\Lezioni\Lezione_VII>gcc -E compilabile.c -o prepro.c
```

# Opzioni del compilatore

```
D:\Codice\Lezioni\Lezione_VII>more < prepro.c
```

```
# 1 "compilabile.c"  
# 1 "<built-in>"  
# 1 "<command line>"  
# 1 "compilabile.c"
```

Contenuto di stdio.h



```
int main ()  
{  
    printf("Hello world!");  
    return(0);  
}
```



# Messaggi del compilatore

## 1. **Messaggi di errore:**

- **indicativi di errori che impediscono la generazione dell'eseguibile**
- **interrompono la compilazione**

## 2. **Warning:**

- **indicativi di condizioni che potrebbero, tra l'altro, inficiare la portabilità del codice**
- **non interrompono la compilazione**

# Messaggi di errore

```
D:\codice\Lezioni\Lezione_VII>type errato.c
// sorgente Lezioni\Lezioni_VII\errato.c
// file non compilabile a causa di errori sintattici
#define STRINGA "Hello world!"
#include <stdio.h>

int main ()
{
    pntf(STRINGA);
    return(0);
}
```

```
D:\codice\Lezioni\Lezione_VII>gcc errato.c
```

```
C:\Users\gaibisso\AppData\Local\Temp\cc2iSB50.o:errato.c:(.text+0x32): undefined reference to `pntf'
collect2: ld returned 1 exit status
```

# Warning

- **Utilizzare il massimo livello di warning:**  
**gcc -Wall nome\_sorgente**

```
// sorgente: Lezione_VII\warning.c
// file compilabile ma con potenziali problemi
// di portabilita'
#define STRINGA "Hello world!"
#include <stdio.h>
main ()
{
    printf(STRINGA);
}
```

```
D:\Codice\Lezioni\Lezione_VII>gcc warning.c
D:\Codice\Lezioni\Lezione_VII>
```

# Warning

- **Utilizzare il massimo livello di warning:**  
**gcc -Wall nome\_sorgente**

```
// sorgente: Lezione_VII\warning.c
// file compilabile ma con potenziali problemi
// di portabilita'
#define STRINGA "Hello world!"
#include <stdio.h>
main () ←
{
    printf(STRINGA);
} ←
```

```
D:\Codice\Lezioni\Lezione_VII>gcc -wall warning.c
warning.c:12: warning: return type defaults to `int'
warning.c: In function `main':
warning.c:14: warning: control reaches end of non-void function
```

# Warning

```
1 // sorgente: Lezione_VII\warning.c
2
3 // file compilabile ma con potenziali problemi
4 // di portabilita'
5
6 #define STRINGA "Hello World!"
7
8 #include <stdio.h>
9
10 int main ()
11 {
12     printf(STRINGA) ;
13 }
14
```

```
D:\codice\Lezioni\Lezione_VII>gcc -Wall warning.c
```

```
warning.c: In function `main':
warning.c:13: warning: control reaches end of non-void function
```

# Warning

```
1 // sorgente: Lezione_VII\warning.c
2
3 // file compilabile ma con potenziali problemi
4 // di portabilita'
5
6 #define STRINGA "Hello World!"
7
8 #include <stdio.h>
9
10 int main ()
11 {
12     printf(STRINGA) ;
13
14     return (0) ;
15 }
```

```
D:\Codice\Lezioni\Lezione_VII>gcc -Wall warning.c
D:\Codice\Lezioni\Lezione_VII>
```