LINGUAGGI FORMALI E TRADUTTORI

LINGUAGGI

LINGUAGGI NATURALI

Italiano, inglese, napoletano..

Flessibili → ricchezza espressiva

GRAMMATICA SINTASSI SEMANTICA

LINGUAGGI FORMALI

Musica, scacchi, matematica, informatica..

Rigorosi → nessuna ambiguità

GRAMMATICA SINTASSI SEMANTICA

ALESSANDRO URSOMANDO

LINGUAGGI FORMALI PER L'INFORMATICA

LINGUAGGI DI BASSO LIVELLO

Machine oriented

Erano difficili da usare
ma avevano prestazioni altissime

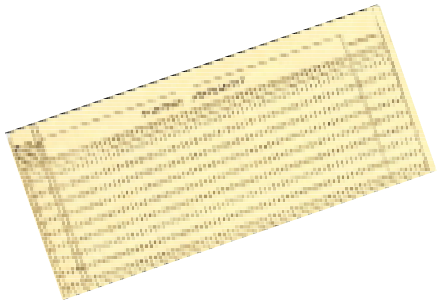
LINGUAGGI DI ALTO LIVELLO

Human oriented

Sono facili da usare
ma il sw è un po' più lento

LINGUAGGI DI BASSO LIVELLO

LINGUAGGI MACCHINA

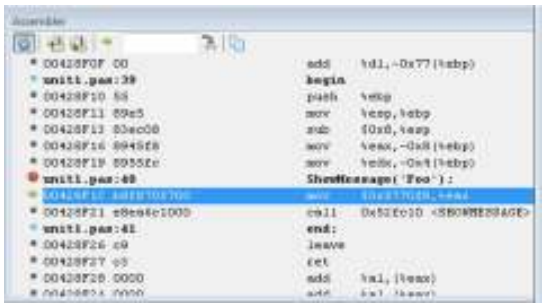


Le istruzioni che la CPU è capace
di eseguire sono sequenze binarie

Scrivere un programma significa scrivere
quelle specifiche sequenze binarie

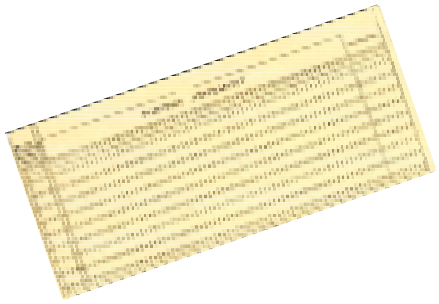
Ovviamente ogni CPU aveva un linguaggio

LINGUAGGI ASSEMBLY



LINGUAGGI DI BASSO LIVELLO

LINGUAGGI MACCHINA



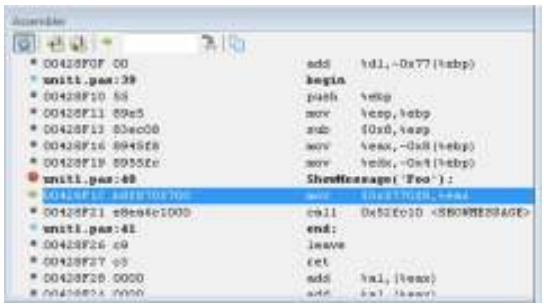
Programmare in questi linguaggi
era molto difficile

1) quel che sapevi fare con una macchina
non andava bene per le altre

2) era molto facile
sbagliare un bit di un'operazione

3) la leggibilità era nulla.

LINGUAGGI ASSEMBLY

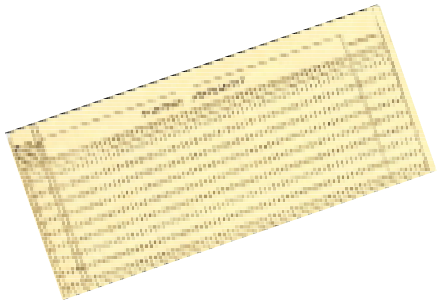


ALESSANDRO URSOMANDO

LINGUAGGI → LINGUAGGI FORMALI → BASSO LIVELLO
CC BY

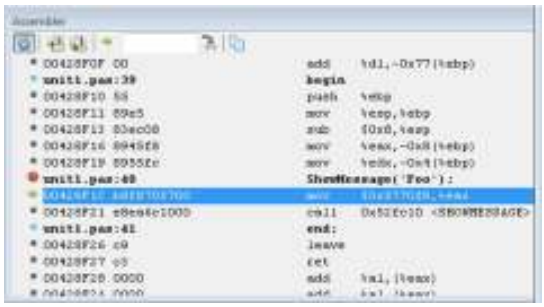
LINGUAGGI DI BASSO LIVELLO

LINGUAGGI MACCHINA



Non c'erano file:
il programma era una serie di schede
perforate e l'esecuzione era immediata

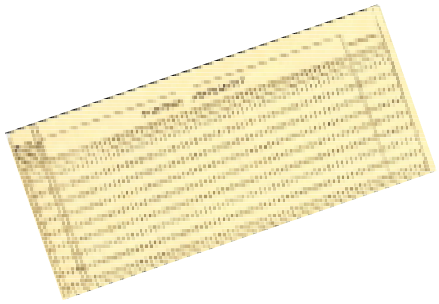
LINGUAGGI ASSEMBLY



ALESSANDRO URSOMANDO

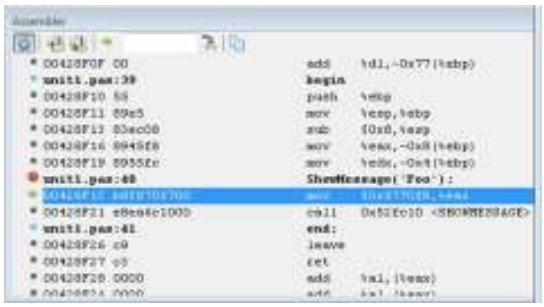
LINGUAGGI DI BASSO LIVELLO

LINGUAGGI MACCHINA



Questi sono i linguaggi di
prima generazione

LINGUAGGI ASSEMBLY

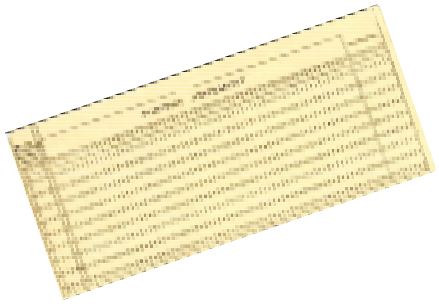


Successivamente sono arrivati
i linguaggi di **seconda generazione:**
i linguaggi assembly

Siamo ancora machine-oriented
ma cerchiamo di facilitare il lavoro del programmatore

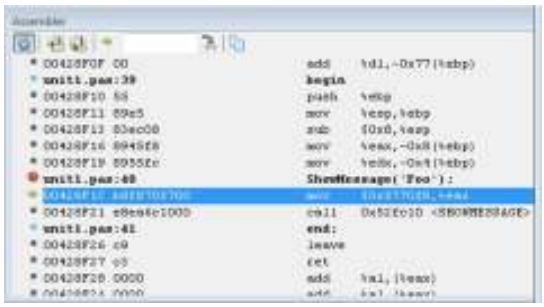
LINGUAGGI DI BASSO LIVELLO

LINGUAGGI MACCHINA



Questi sono i linguaggi di
prima generazione

LINGUAGGI ASSEMBLY



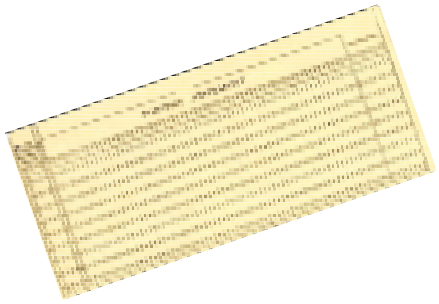
Per ogni istruzione che la CPU è capace di eseguire c'è un'istruzione assembly.

Le istruzioni sono parole mnemoniche
MOV $1, 10

Evidentemente ogni CPU ha il suo assembly.

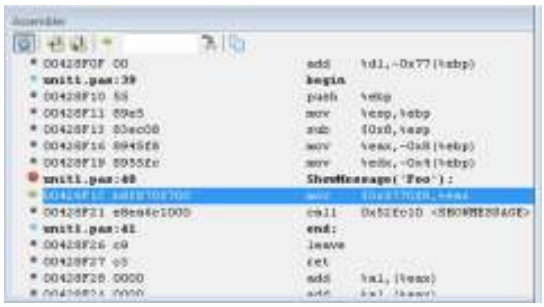
LINGUAGGI DI BASSO LIVELLO

LINGUAGGI MACCHINA



Questi sono i linguaggi di
prima generazione

LINGUAGGI ASSEMBLY

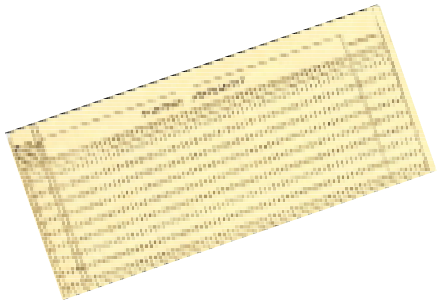


Il programmatore
scrive un file con le istruzioni assembly
e c'è poi un software che traduce quelle
istruzioni e che produce un file eseguibile

ALESSANDRO URSOMANDO

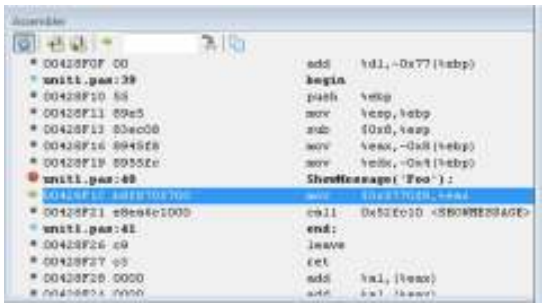
LINGUAGGI DI BASSO LIVELLO

LINGUAGGI MACCHINA



Questi sono i linguaggi di
prima generazione

LINGUAGGI ASSEMBLY

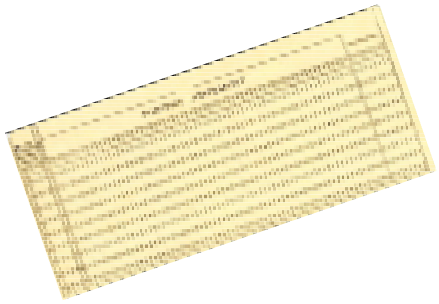


Questo traduttore si chiama
assembler

ALESSANDRO URSOMANDO

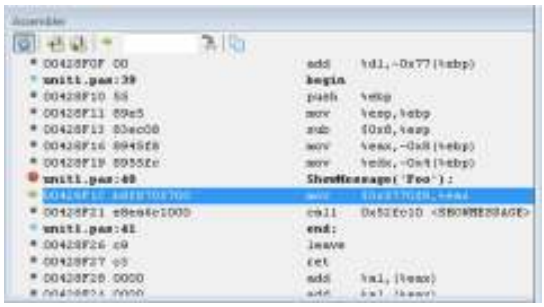
LINGUAGGI DI BASSO LIVELLO

LINGUAGGI MACCHINA

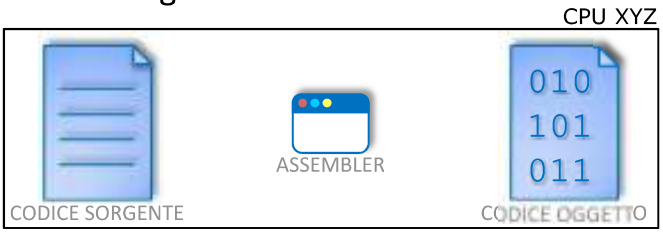


Questi sono i linguaggi di
prima generazione

LINGUAGGI ASSEMBLY

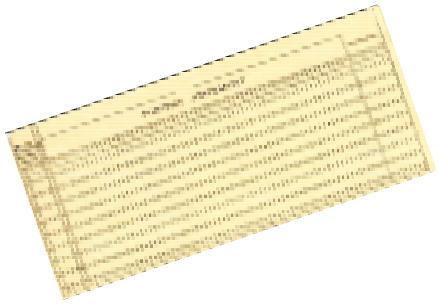


Ovviamente ogni CPU
ha il suo assembly e il suo assembler
Ovviamente ogni assembler produce codici
eseguibili solo dalla sua CPU



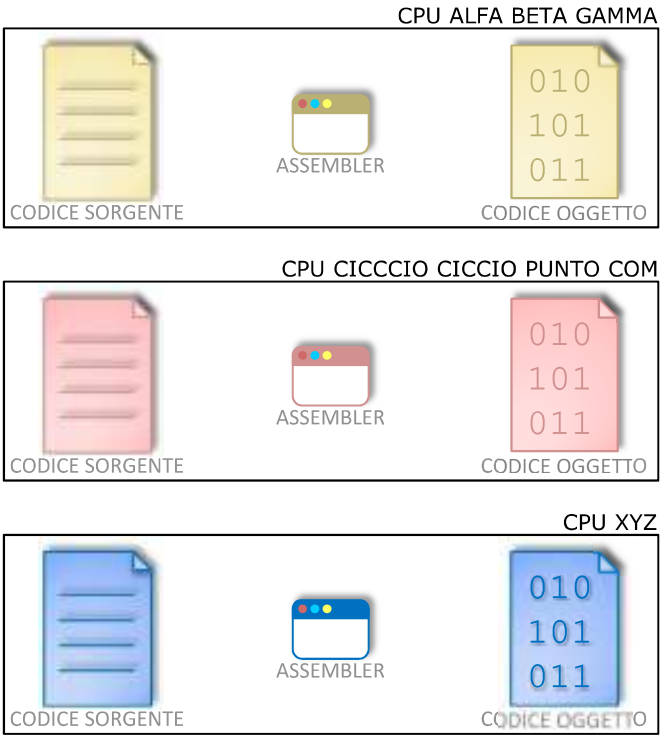
LINGUAGGI DI BASSO LIVELLO

LINGUAGGI MACCHINA



Questi sono i linguaggi di
prima generazione

LINGUAGGI ASSEMBLY



LINGUAGGI DI TERZA GENERAZIONE O LINGUAGGI DI ALTO LIVELLO

Sono i primi linguaggi human-oriented

Fortran, Cobol, C, Pascal..

C++, Java, PHP, Python..

USABILITÀ

PORTABILITÀ

I linguaggi di terza generazione nascono con l'idea di spingere sul concetto di «traduttore».

e questo conferisce loro due caratteristiche fondamentali

LINGUAGGI DI TERZA GENERAZIONE O LINGUAGGI DI ALTO LIVELLO

Sono i primi linguaggi human-oriented

Fortran, Cobol, C, Pascal..

C++, Java, PHP, Python..

USABILITÀ

PORTABILITÀ

USABILITÀ

L'usabilità misura il livello di facilità d'uso di uno strumento tecnologico

LINGUAGGI DI TERZA GENERAZIONE O LINGUAGGI DI ALTO LIVELLO

Sono i primi linguaggi human-oriented

Fortran, Cobol, C, Pascal..

C++, Java, PHP, Python..

USABILITÀ

PORTABILITÀ

PORTABILITÀ

La portabilità misura la possibilità di usare uno strumento tecnologico in ambiti diversi

LINGUAGGI DI TERZA GENERAZIONE O LINGUAGGI DI ALTO LIVELLO

Sono i primi linguaggi human-oriented

Fortran, Cobol, C, Pascal..

C++, Java, PHP, Python..

USABILITÀ

PORTABILITÀ

USABILITÀ

Semplifico il linguaggio con cui scrivo il codice sorgente

```
mov ax,0
mov ax,cx
out 70,al
mov ax,0
out 71,al
inc cx
mov ax,0
mov ax,cx
out 70,al
mov ax,0
out 71,al
inc cx
mov ax,0
mov ax,cx
out 70,al
mov ax,0
out 71,al
inc cx
```

CODICE SORGENTE

CODICE SORGENTE

LINGUAGGI DI TERZA GENERAZIONE O LINGUAGGI DI ALTO LIVELLO

Sono i primi linguaggi human-oriented

Fortran, Cobol, C, Pascal..

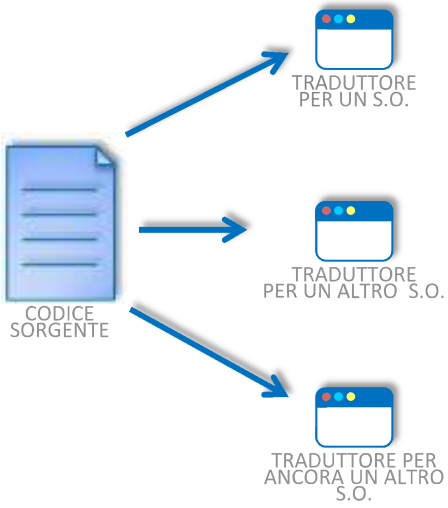
C++, Java, PHP, Python..

USABILITÀ

PORTABILITÀ

PORTABILITÀ

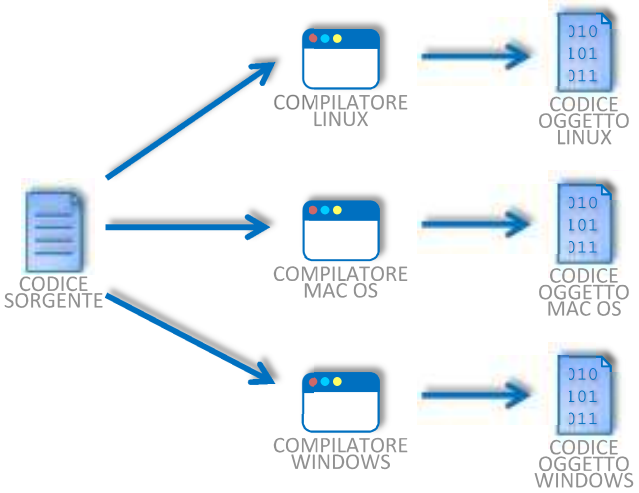
Invece di tradurre direttamente per una specifica CPU mi conviene usare un livello di astrazione più alto e produrre codice per uno specifico sistema operativo.



ALESSANDRO URSOMANDO

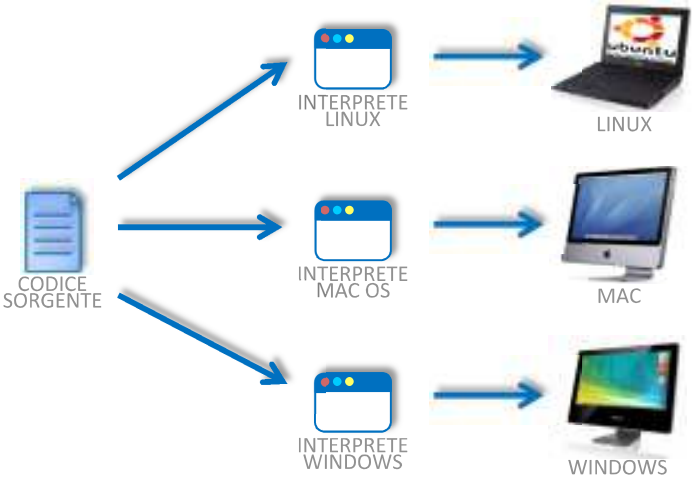
LINGUAGGI DI TERZA GENERAZIONE O LINGUAGGI DI ALTO LIVELLO

LINGUAGGI COMPILATI



Il compilatore ha come obiettivo la realizzazione di un file oggetto che poi verrà eseguito dal sistema operativo.

LINGUAGGI INTERPRETATI

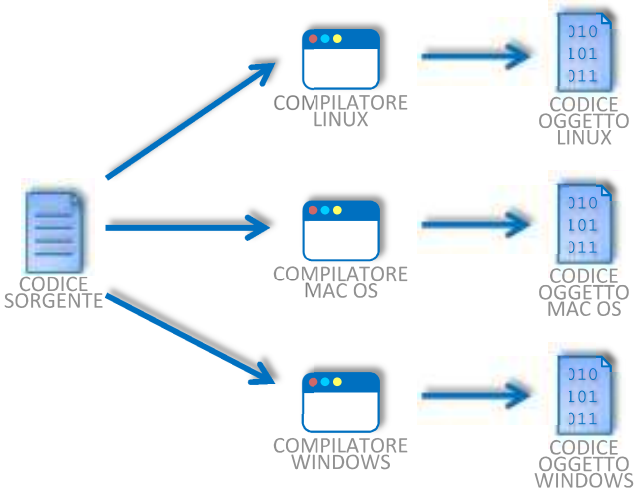


L'interprete traduce ed esegue al volo il codice sorgente una istruzione alla volta.

ALESSANDRO URSOMANDO

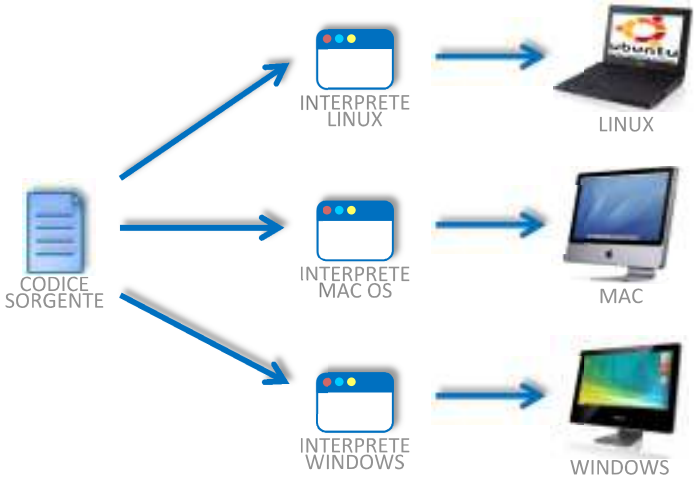
LINGUAGGI DI TERZA GENERAZIONE O LINGUAGGI DI ALTO LIVELLO

LINGUAGGI COMPILATI



C, C++, Pascal, …

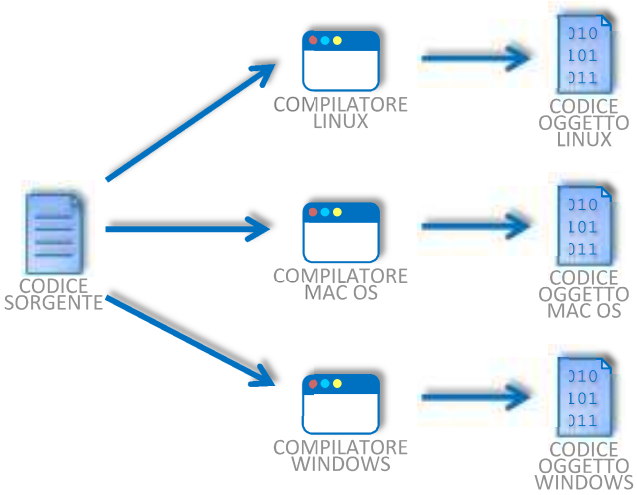
LINGUAGGI INTERPRETATI



PHP, Perl, Visual Basic, Javascript, …

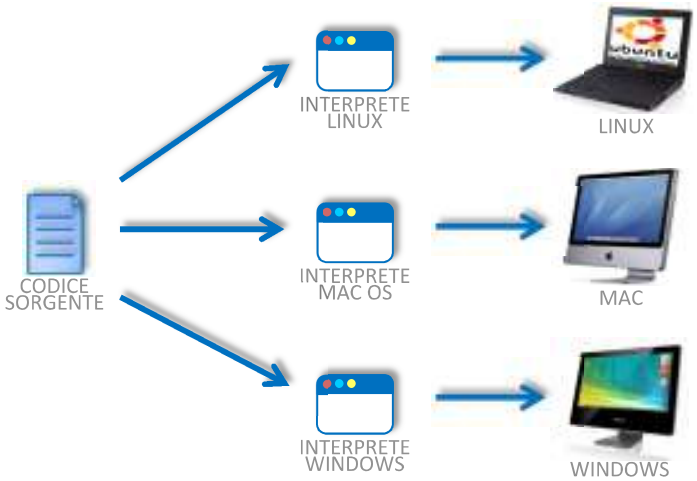
LINGUAGGI DI TERZA GENERAZIONE O LINGUAGGI DI ALTO LIVELLO

LINGUAGGI COMPILATI



Se c'è un errore nel codice sorgente, il compilatore non riesce a creare il file oggetto (e segnala l'errore).

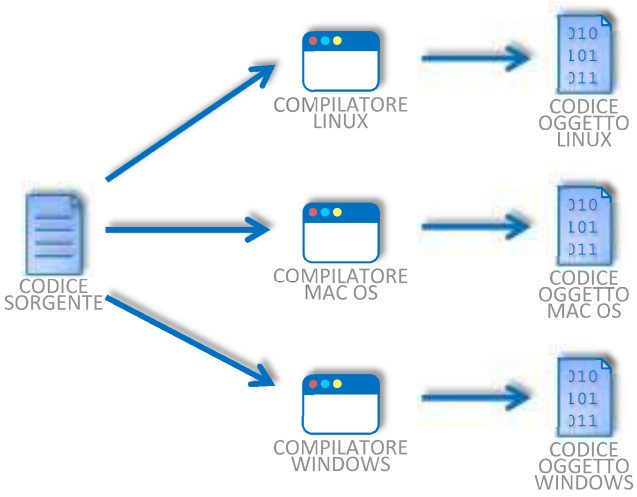
LINGUAGGI INTERPRETATI



L'interprete invece esegue tutte le istruzioni corrette fino all'errore.
Dopo l'errore alcuni interpreti interrompono l'esecuzione altri invece procedono ad eseguire le istruzioni successive

LINGUAGGI DI TERZA GENERAZIONE O LINGUAGGI DI ALTO LIVELLO

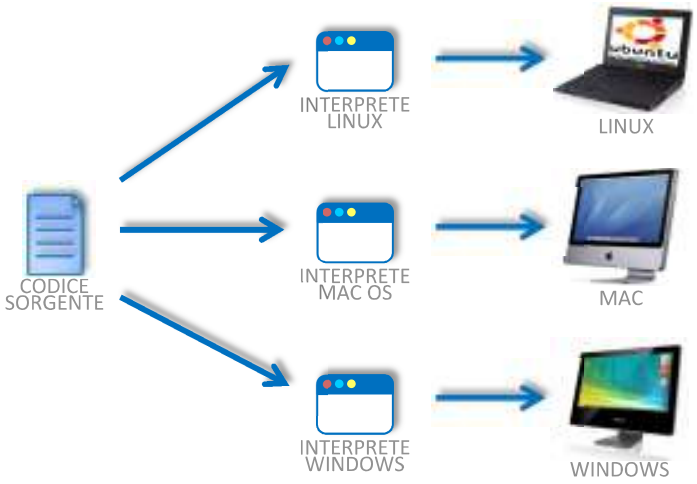
LINGUAGGI COMPILATI



Sviluppando in un linguaggio compilato non consegniamo il codice sorgente al cliente.

Il cliente necessita solo del computer per eseguire il programma.

LINGUAGGI INTERPRETATI

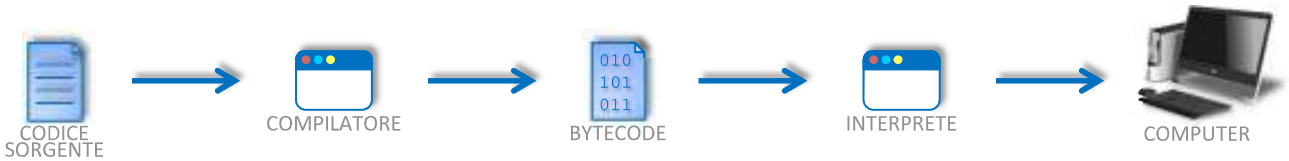


Sviluppando in un linguaggio interpretato dobbiamo dare il codice sorgente al cliente.

Il cliente necessita del computer e dell'interprete per eseguire il programma

LINGUAGGI DI TERZA GENERAZIONE O LINGUAGGI DI ALTO LIVELLO

LINGUAGGI SEMICOMPILATI



Java, Python, ..

Il bytecode è un file oggetto scritto con le istruzioni di una macchina che non esiste

Il bytecode è eseguibile solo in una speciale macchina virtuale (l'interprete)

Come in un linguaggio compilato non è necessario consegnare il codice sorgente all'utente finale

Come in un linguaggio interpretato ho una portabilità molto alta

L'esecuzione è più veloce di quella dei linguaggi interpretati

I TEMPI E GLI ERRORI DELLA PROGRAMMAZIONE

DESIGN TIME

Quando scrivo codice sono a design-time

RUN TIME

Quando eseguo il codice sono a run-time

I TEMPI E GLI ERRORI DELLA PROGRAMMAZIONE

DESIGN TIME

Gli errori che posso individuare
a **design-time** sono gli errori **lessicali**
(il compilatore non riconosce un simbolo),
sintattici (manca il ; alla fine), **semantici**
(variabili non dichiarate, type checking, ..)

RUN TIME

A **run-time** invece posso verificare la
presenza di errori **logici** o di **run-time**
if (a=0)
div by zero, overflow

I SW PER PROGRAMMARE

Per sviluppare sw con un linguaggio interpretato servono:

Un editor di testo

e l'interprete per la nostra piattaforma

I SW PER PROGRAMMARE

Per sviluppare sw con un linguaggio compilato servono:

Un editor di testo

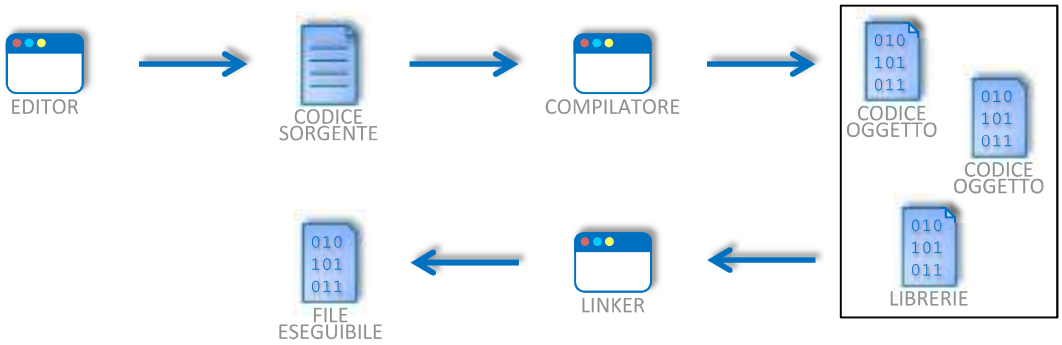
I SW PER PROGRAMMARE

Per sviluppare sw con un linguaggio compilato servono:

Il compilatore (per la macchina di riferimento)

I SW PER PROGRAMMARE

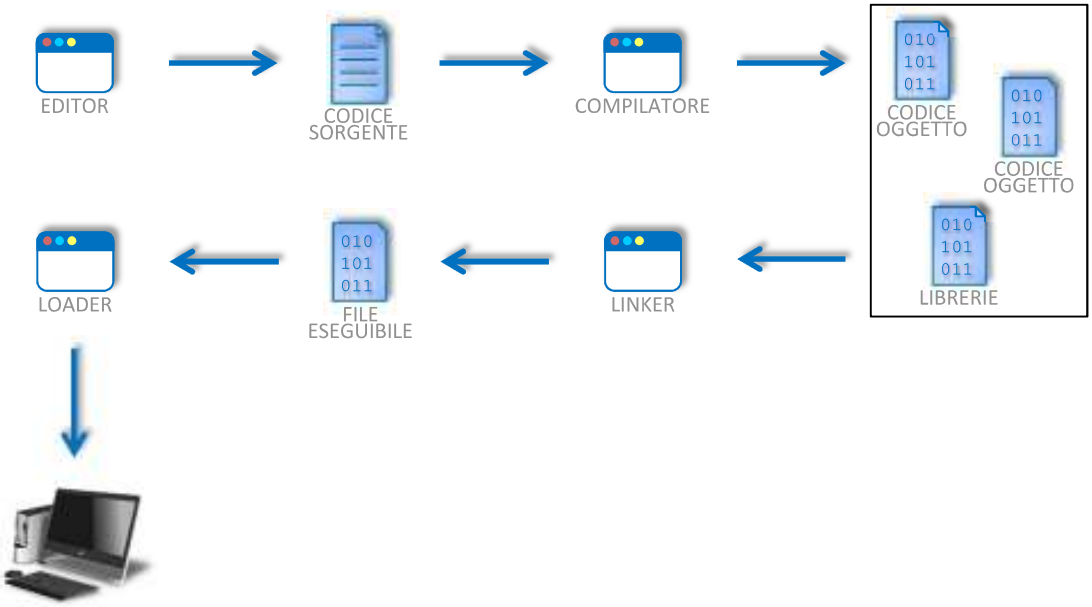
Per sviluppare sw con un linguaggio compilato servono:



Se il progetto è molto grande il codice oggetto di ciascun modulo e le librerie usate vanno linkate insieme per produrre il file eseguibile.

I SW PER PROGRAMMARE

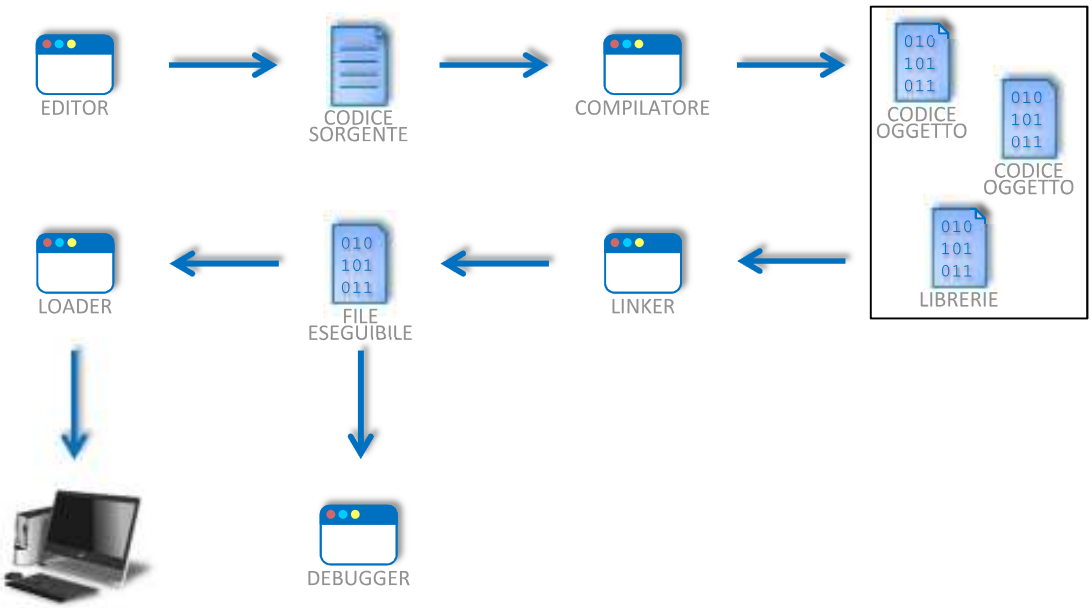
Per sviluppare sw con un linguaggio compilato servono:



Il file eseguibile è caricato in memoria da un modulo del sistema operativo che si chiama loader ed il relativo processo eseguito dallo scheduler

I SW PER PROGRAMMARE

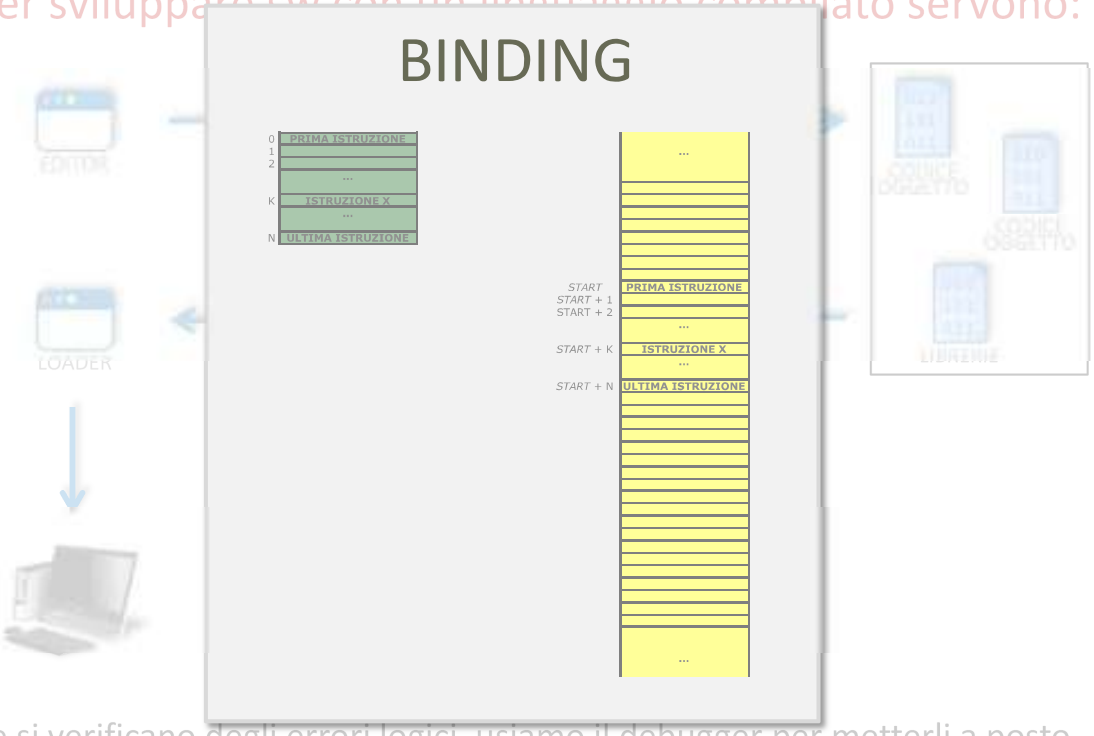
Per sviluppare sw con un linguaggio compilato servono:



Se si verificano degli errori logici, usiamo il debugger per metterli a posto.

I SW PER PROGRAMMARE

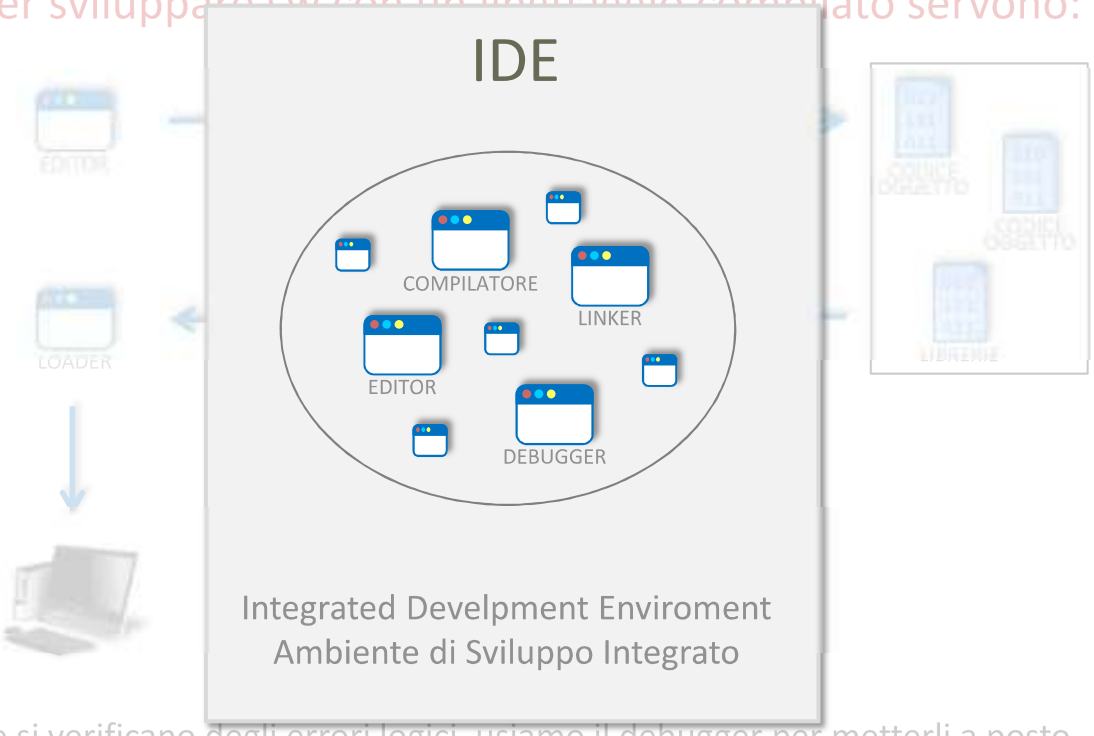
Per sviluppare sw con un linguaggio compilato servono:



Se si verificano degli errori logici, usiamo il debugger per metterli a posto.

I SW PER PROGRAMMARE

Per sviluppare sw con un linguaggio compilato servono:



Se si verificano degli errori logici, usiamo il debugger per metterli a posto.

I LINGUAGGI DI QUARTA E QUINTA GENERAZIONE

QUARTA GENERAZIONE

CSS, SQL, fogli elettronici..

QUINTA GENERAZIONE

I linguaggi destinati ad applicazioni di intelligenza artificiale

ALESSANDRO URSOMANDO

I PARADIGMI DI PROGRAMMAZIONE

INTRODUZIONE

Ogni linguaggio appartiene a un certo paradigma di programmazione

Un paradigma di programmazione rappresenta quindi una serie di caratteristiche comuni a diversi linguaggi

In ambito scientifico con «cambio di paradigma» si intende il sovvertire gli assunti di base di una teoria universamente accettata.

Un paradigma di programmazione è un modo di pensare la programmazione.

DEFINIZIONE

PRIMA SUDDIVISIONE: L'OBIETTIVO DELL'ISTRUZIONE

PARADIGMA DICHIARATIVO

Ogni istruzione descrive il risultato che si vuole ottenere.

```
<h1>Hello world</h1>
```

PARADIGMA IMPERATIVO

Ogni istruzione dice cosa fare per ottenere il risultato.

```
print("Hello World")
```

SOTTO - PARADIGMI

PARADIGMA DICHIARATIVO

Ogni istruzione descrive
il risultato che si vuole ottenere.

<h1>Hello world</h1>

PARADIGMA IMPERATIVO

Ogni istruzione dice cosa fare
per ottenere il risultato.

print("Hello World")

PARADIGMA DICHIARATIVO

PARADIGMA DICHIARATIVO

- Funzionale
- Logico
- A vincoli

Ogni istruzione descrive
il risultato che si vuole ottenere.

<h1>Hello world</h1>

In generale, poco diffuso

Html e **query language**
sono i linguaggi più fortunati

Python, **Javascript**
e altri linguaggi famosi
supportano anche il paradigma funzionale
anche se raramente vengono usati così

Esistono infine alcuni linguaggi
funzionali che, sebbene non
siano molto diffusi, sono alla base di alcuni
tra i software più utilizzati al mondo.
Whatsapp (Erlang), il 90% degli **switch**
(Erlang), Messanger di **Facebook**
(ReasonML), **Firefox** (Rust).

PARADIGMA IMPERATIVO

	Procedurale	Strutturato	OOP
	Primi anni 60	Poco dopo	nasce nel 67 ma domina gli anni 90
	Fortran, Cobol	C, Pascal	C++, Java, Python
	Input Output Assegnazione Iterazione Selezione Subroutine Goto	Input Output Assegnazione Iterazione Selezione Subroutine No Goto + altri accorgimenti	Tutto quel che c'era nella programmazione strutturata + classi, oggetti, metodi e attributi

SOFTWARE:
CONCETTI DI BASE

CLASSIFICAZIONE PER OBIETTIVO

SW DI BASE

Tutto il sw che permette di utilizzare e gestire l'hardware.

Sistemi Operativi, driver, utility.
Sw per la creazione di altri sw.

SW APPLICATIVO

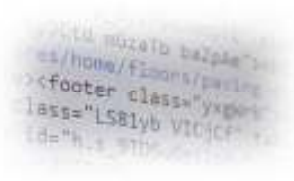
Tutti i sw che permettono l'automazione di una qualsiasi attività.

Office Automation.
Videogiochi.
Gestionali.
Ecc.

CLASSIFICAZIONE PER OBIETTIVO

SW APPLICATIVO

SOFTWARE
PROGRAMMA
APPLICAZIONE



Sono sinonimi.
Indicano un sw generico, di qualsiasi tipo.

WEB-APP

E' un'applicazione che viene eseguita all'interno di un browser

APP

Le «app» nascono con lo smartphone.

L'app tipicamente è scaricabile da uno store.

(ma non per forza)

CLASSIFICAZIONE PER OBIETTIVO

SW APPLICATIVO

SOFTWARE PROGRAMMA APPLICAZIONE

Sono sinonimi.

Indicano un sw generico, di qualsiasi tipo.

WEB-APP

E' un'applicazione che viene eseguita all'interno di un browser

APP

Windows 10 ha importato il concetto di store e quindi quello di app.

Pertanto la app non è più legata al concetto di smartphone.

CLASSIFICAZIONE PER TIPOLOGIA DI CLIENTE

CLIENTE ≠ UTENTE

CLIENTE

La persona o l'azienda che paga per il software

UTENTE

Una delle persone che usa il software

CLASSIFICAZIONE PER TIPOLOGIA DI CLIENTE

SW ORIZZONTALE

GENERAL POURPOSE

Software realizzato da un'azienda immaginando le esigenze del pubblico e poi immesso sul mercato

SW VERTICALE

CUSTOM

Software realizzato su commessa che risolve le esigenze di una specifica realtà

CLASSIFICAZIONE PER DISPONIBILITÀ DEL CODICE

SW LIBERO OPEN SOURCE

Il produttore rende pubblico il codice sorgente. Chiunque può sviluppare sw a partire da esso ma anche questo nuovo sw sarà open source.

SW DI PUBBLICO DOMINIO

Il produttore rinuncia alla paternità del sw. Chiunque può sviluppare sw a partire da esso e questo nuovo sw può anche non essere di pubblico dominio.

SW PROPRIETARIO

Il produttore tiene per sé il codice sorgente.

CLASSIFICAZIONE PER PAGAMENTO

FREEWARE

E’ gratuito

SW COMMERCIALE

Ha un costo

SHAREWARE

E’ un sw gratuito che spinge all’acquisto di una versione a pagamento (più completa) dello stesso sw.

COMBINAZIONE DEI DUE ASPETTI

COMBINAZIONE DEI DUE ASPETTI

COMBINAZIONE DEI DUE ASPETTI

CICLO DI VITA DEL SOFTWARE

CICLO DI VITA DEL SOFTWARE

- Analisi del problema

CICLO DI VITA DEL SOFTWARE

CICLO DI VITA DEL SOFTWARE

- Analisi del problema
- Progettazione del software

CICLO DI VITA DEL SOFTWARE

PROGETTISTA

CICLO DI VITA DEL SOFTWARE

HARDWARE

- Macchine
- Dispositivi
- Connessioni

PROGETTISTA

SOFTWARE

- Linguaggi
- Tecnologie
- Framework

RISORSE UMANE

- Capacità
- Conoscenze
- Anzianità

CICLO DI VITA DEL SOFTWARE

- Analisi del problema
- Progettazione del software
- Implementazione del software

CICLO DI VITA DEL SOFTWARE

CODING TEAM

CICLO DI VITA DEL SOFTWARE

- Analisi del problema
- Progettazione del software
- Implementazione del software
- Beta Test (e dov'è l'alfa test ?)

CICLO DI VITA DEL SOFTWARE

BETA TESTER

CICLO DI VITA DEL SOFTWARE

- Analisi del problema
- Progettazione del software
- Implementazione del software
- Beta Test (e dov'è l'alfa test ?)
- Rilascio: promozione / setup e training

CICLO DI VITA DEL SOFTWARE

PROMOZIONE

SETUP E TRAINING

CICLO DI VITA DEL SOFTWARE

- Analisi del problema
- Progettazione del software
- Implementazione del software
- Beta Test (e dov'è l'alfa test ?)
- Rilascio: promozione / setup e training
- Manutenzione: supporto, correttiva, evolutiva

CICLO DI VITA DEL SOFTWARE

SUPPORTO

CORRETTIVA

EVOLUTIVA