

# **LABORATORIO DI INFORMATICA**

**Lezione 24/10/2019**

Il **software**, in informatica e in elettronica, è l'insieme delle componenti immateriali (strato logico/intangibile) di un sistema elettronico di elaborazione, in contrapposizione all'hardware, cioè la parte materiale (strato fisico/tangibile) dello stesso sistema

Per **software di base**, o di sistema, si intendono tutti i programmi basilari per il funzionamento dell'elaboratore e di tutti i suoi componenti.

Il **sistema operativo** costituisce la parte fondamentale del software di base, in quanto coordina il collegamento tra le varie componenti del computer, come vedremo in maniera più approfondita parlando dei sistemi operativi.

Il **software applicativo**, invece, è composto da tutti i programmi che risolvono i problemi specifici dell'utente, rispondendo ad esigenze di varia natura:

- *elaborazione di testi*: scrittura, controllo ortografico, stampa ecc.;
- *foglio elettronico* (spreadsheet): calcoli, grafici, statistiche;
- *gestione di basi di dati*: organizzazione dei dati;
- *disegno*: gestione di immagini, ritocco fotografico, grafica;
- *gestione audio*: creazione di suoni, registrazione ecc.;
- *animazione video*: creazione o modifica di filmati;
- *programmazione*: linguaggio C, Java, Html ecc.



# SISTEMA OPERATIVO

Definizione di S.O.

Si potrebbe adottare quella per cui il S.O. è quel programma sempre in esecuzione, il kernel (nucleo) cioè il cuore del sistema software della macchina

- il programma che agisce come intermediario tra l'hardware della macchina e l'utente del computer

Gli obiettivi del S.O. sono:

- eseguire i programmi utente e semplificare la soluzione dei “problemi” dell'utente
- rendere il “sistema computer” agevole da utilizzare
- sfruttare l'hardware del computer nella maniera più efficiente possibile
- controllare e coordinare l'uso dell'hardware da parte dei diversi applicativi dei diversi utenti



Il kernel è il componente centrale di ogni sistema operativo. Fa da ponte tra le componenti hardware di un computer – come processore, RAM e hard disk – e i programmi in esecuzione sul computer stesso. Data la sua rilevanza, è il primo programma ad essere caricato in memoria quando si accende un computer e l'ultimo ad essere chiuso in fase di spegnimento.

Il kernel è responsabile della gestione e dell'allocazione delle risorse del computer, permettendo ad altri applicazioni e programmi di utilizzare queste risorse. Definisce, inoltre, lo spazio degli indirizzi di memoria per i vari programmi in esecuzione, carica in memoria i file con il codice dell'applicazione e imposta l'ordine di esecuzione delle varie applicazioni e dei programmi attivi. **I compiti principali di un kernel, quindi, possono essere raggruppati in quattro categorie principali: gestione dei processi per l'esecuzione dei programmi, gestione e allocazione della memoria e gestione della comunicazione input e output del computer, gestioni delle componenti hardware e delle periferiche.**



Lo scheduler è il componente fondamentale dei sistemi operativi multitasking, cioè quelli in grado di eseguire più processi contemporaneamente (va inteso simbolicamente. In realtà non c'è una gestione parallela dei processi ma avviene in sequenza, i tempi sono talmente brevi che all'utente sembrerà che i programmi vadano contemporaneamente). Lo scheduler si occupa di fare avanzare un processo interrompendone temporaneamente un altro, realizzando così un cambiamento di contesto (*context switch*). Generalmente computer con un processore sono in grado di eseguire un programma per volta, quindi per poter far convivere più task è necessario usare lo scheduler. Esistono vari algoritmi di scheduling che permettono di scegliere nella maniera più efficiente possibile quale processo far proseguire.



## Definizione:

**Un sistema operativo è un programma che controlla l'esecuzione di programmi applicativi e agisce come interfaccia tra le applicazioni e l'hardware del calcolatore**

## Obiettivi

Efficienza: Un S.O. cerca di utilizzare in modo efficiente le risorse del calcolatore

Semplicità: Un sistema operativo dovrebbe semplificare l'utilizzazione dell'hardware di un calcolatore

### Esempio senza S.O.

```
li $t0, 0xDEFF12 # init
sw $t0, 0xB000.0040
li $t0, 0xFFDF # motor
sw $t0, 0xB000.0044
li $t0, 0xFFBB
sw $t0, 0xB000.0048
...
```



### Esempio: floppy disk drive

inizializzazione, avviamento motore, spostamento testina, lettura scrittura, spegnimento motore



## **Servizi estesi offerti da un S.O.:**

**esecuzione di programmi**

**accesso semplificato ai**

**dispositivi di I/O**

**rilevazione e risposta agli errori**

**accounting**

**L'evoluzione dei sistemi operativi è stata spinta dal progresso tecnologico nel campo dell'hardware ed ha guidato il progresso tecnologico nel campo dell'hardware**



## Storia dei Sistemi Operativi

**Generazione 1: 1945 - 1955** valvole e tavole di commutazione

**Generazione 2: 1955 - 1965** transistor e sistemi batch

**Generazione 3: 1965 – 1980** circuiti integrati,  
multiprogrammazione e time-sharing

**Generazione 4: 1980 – oggi** personal computer





## Generazione 1 (1944-1955)

Come venivano costruiti?

macchine a valvole e tavole di commutazione

Come venivano usati?

solo calcoli numerici (calcolatori non elaboratori) **un singolo gruppo di persone progettava, costruiva, programmava e manteneva il proprio computer**

Come venivano programmati?

in linguaggio macchina

programmazione su tavole di commutazione

**Nessun sistema operativo!**



## Principali problemi

**grossi problemi di affidabilità** (guasti frequenti)

**rigidità nell'assegnazione dei ruoli;**

non esiste il concetto di programmatore come entità separata dal costruttore di computer e dall'utente

**utilizzazione lenta e complessa;**

l'operatore doveva:

caricare il programma da eseguire

inserire i dati di input eseguire il programma attendere il risultato ricominciare dal punto 1.

**Tutto ciò a causa dell'assenza del sistema operativo**



## **Generazione 2 (1955-1965)**

### **Come venivano costruiti?**

introduzione dei transistor

costruzione di macchine più affidabili ed economiche

### **Come venivano usati?**

le macchine iniziano ad essere utilizzate per compiti diversi  
si crea un mercato, grazie alle ridotte dimensioni e al prezzo più  
abbordabile

avviene una separazione tra costruttori, operatori e  
programmatori

### **Come venivano programmati?**

linguaggi ad "alto livello": Assembly, Fortran  
tramite schede perforate

### **Sistemi operativi batch**



## Definizione: job

Un programma o un'insieme di programmi la cui esecuzione veniva richiesta da uno degli utilizzatori del computer

Ciclo di esecuzione di un job:

il programmatore :

scrive (su carta) un programma in un linguaggio ad alto livello  
perfora una serie di schede con il programma e il suo input  
consegna le schede ad un operatore

l'operatore :

inserisce le schede del programma  
attende il risultato e lo consegna al programmatore

**Nota: operatore != programmatore == utente**



Il termine *batch* (letteralmente: infornata) viene usato per indicare un particolare modo di gestire le operazioni di un sistema di calcolo.

In sostanza significa che tutta la sequenza di operazioni e dati necessari per svolgere un particolare compito vengono preparati in anticipo e memorizzati su un adeguato supporto creando una *procedura*. La procedura viene poi eseguita dal sistema come un blocco unico, cioè senza che sia necessario, o possibile, un intervento umano prima che sia terminata.

Il termine è contrapposto alla modalità *interattiva*, che si ha quando l'utente accede direttamente ad un terminale o ad una interfaccia dell'elaboratore e segue passo passo le varie fasi dell'operazione.

Nei primi sistemi di calcolo la modalità batch era l'unica disponibile: l'utente preparava in anticipo un "pacchetto" di schede perforate che contenevano le istruzioni per il sistema operativo, il codice del programma ed i dati di ingresso. Il pacchetto di schede veniva poi consegnato ad un operatore che provvedeva ad inserirlo in una coda di esecuzione. Il risultato era costituito normalmente da una stampa che veniva riconsegnata all'utente al termine del lavoro.



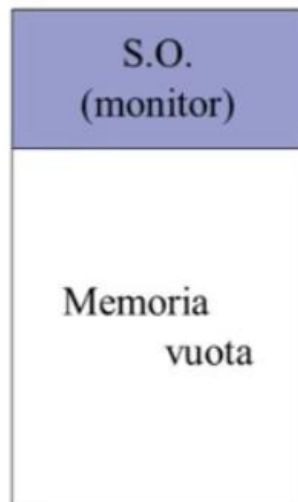
La modalità batch aveva il vantaggio di consentire un uso continuativo del sistema di calcolo, dato che si faceva in modo che nella coda di esecuzione ci fossero sempre procedure in attesa di essere eseguite, ed inoltre una gestione di questo tipo non richiede sistemi operativi particolarmente complessi. Ciò era particolarmente importante dato che gli elaboratori da un lato disponevano di risorse (velocità e memoria) assai limitate e, dall'altro, avevano costi di acquisto e gestione elevatissimi.



## Principali problemi

Molte risorse restavano inutilizzate:  
durante le operazioni di lettura schede / stampa,  
durante il caricamento di un nuovo job, il  
processore restava inutilizzato  
parte della memoria restava inutilizzata

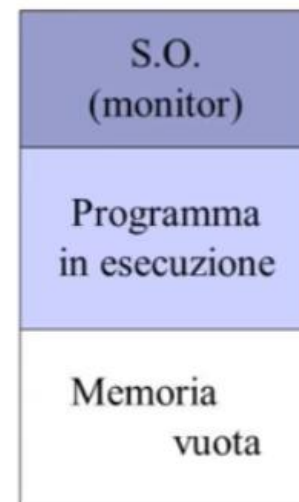
### Memory layout per un S.O. batch (versione semplificata)



1. Stato iniziale



2. Dopo il caricamento del compilatore fortran



3. Dopo la compilazione



## **Generazione 3 (1965-1980)**

### **Come venivano costruiti?**

circuiti integrati

### **Come venivano usati?**

man mano sparisce la figura dell'operatore come "interfaccia" degli utenti verso la macchine  
utente == operatore

### **Come venivano programmati?**

linguaggi ad "alto livello": C, shell scripting  
editor testuali, editor grafici, compilatori  
accesso al sistema da terminali

**Quale sistemi operativi venivano usati?**  
**non più batch ma interattivi**  
**multi-programmazione**  
**time sharing**





## Definizione: multiprogrammazione

Utilizzare il processore durante i periodi di I/O di un job per eseguire altri job

### Vantaggi

il processore non viene lasciato inattivo (idle) durante operazioni di I/O molto lunghe

la memoria viene utilizzata al meglio, caricando il maggior numero di job possibili

Caratteristiche tecniche: Più job contemporaneamente in memoria

Una componente del S.O. detto scheduler si preoccupa di alternarli nell'uso della CPU quando un job richiede un'operazione di I/O, la CPU viene assegnata ad un altro job.

S.O.
Job 1
Job 2
Job 3
Job 4



## Definizione: time shering

Il Time Sharing (a partizione di tempo) è un modo di gestione che assegna a una risorsa un tempo limite per il suo uso, questo periodo di tempo viene detto Time Slice. Quando il time slice termina il sistema operativo “parcheggia” il programma e manda in esecuzione il successivo. Nel sistema operativo vi possono essere dei meccanismi di priorità che permettono ad alcuni programmi di essere eseguiti prima e magari con un time slice più lungo. Va comunque notato che non tutte le periferiche sono gestite dal time sharing, per esempio per una stampante non c'è un time slice in quanto ogni utente usa il dispositivo finché gli serve e poi lo rilascia



## Generazione 4 (1980- )

### Personal computer

Punti chiave

personal computer sono dedicati a singoli utenti

L'obiettivo primario diventa la facilità d'uso  
Essendo dedicati a singoli utenti, i sistemi operativi per PC sono in generale più semplici

Tecnologie a finestre

Negli anni '80 nacquero i primi personal computer. Uno tra i più diffusi fu il PC IBM, con il quale venne introdotto anche il sistema operativo MS-DOS. Sempre in quegli anni la Apple (prendendo spunto da un'idea della Xerox) introdusse la prima **interfaccia grafica** (GUI). L'idea fu subito applicata anche dalla Microsoft con l'introduzione del sistema operativo Windows.



L'MS-DOS è un sistema operativo nato nel 1981 per il PC IBM e adottato in seguito dai suoi successori più evoluti nonché dai numerosissimi compatibili delle più diverse marche. Mentre “MS” indica MicroSoft, “DOS” sta per Disk Operative System ossia sistema operativo a dischetti (il nome deriva dal fatto che può essere contenuto in un numero esiguo di dischetti).

L'MS-DOS è stato lo standard di fatto dei sistemi operativi per computer per più di quasi un ventennio. Fino all'avvento di Windows '95 e successive versioni, sempre della MicroSoft. In tali nuovi sistemi operativi viene mantenuto un ambiente MS-DOS per consentire l'utilizzo di applicativi e comandi usuali del DOS.



# IL PROMPT DEI COMANDI

- MS-DOS è l'acronimo di "Microsoft - Disk Operating System“
- Il primo di una lunga serie di sistemi operativi della Microsoft
- E' un sistema operativo commercializzato dal giugno 1982 al 2001
- Fino alla commercializzazione di Microsoft Windows 95, MS-DOS è stato il sistema operativo per personal computer più diffuso al mondo.



MSDOS (*Microsoft Disk Operating System*) è nato per i computer della serie IBM (e compatibili). Questi computer sono basati sui i micro-processorii prodotti da Intel.

L' IBM PC fù prodotto nel 1981, aveva 640K di RAM, un interfaccia a cassette ed il linguaggio BASIC in ROM, il microprocessore era un 8088 dell'Intel, e la memoria era limitata a 1Mbyte, di cui 640K di RAM, e gli altri riservati per le periferiche hardware come schede grafiche ed espansioni. Il processore 8088 era ad 8 bit processor, il suo clock era a 4.77Mhz.

Insieme con l'IBM PC fu introdotto l'MSDOS (prodotto dalla Microsoft, ed ispirato al CP/M della Digital), che ancora non supportava la struttura gerarchica delle directory.

Nel 1984 IBM inizia a commercializzare l'IBM AT (Advanced Technology), basato sul Intel 80286 a 16 bit, ad 8 MHz. Nel 1987 escono i primi PC basati sull' Intel 80386 (prodotti da Compaq) a 32 bit. MSDOS, seppur notevolmente aggiornato, presenta i suoi limiti, in particolare per la gestione della memoria e la mancanza di un interfaccia grafica. Su macchine basate sul 80386 (e sui successivi 80486 e processori della famiglia Pentium) possono girare altri sistemi operativi: Windows 3.1 (sempre basato su MS-DOS), OS-2 ed Unix. A fine 1994 esce un Unix free per i computer con processori Intel (LINUX), nel 1995 Microsoft rilascia un nuovo sistema operativo a 32 bit completamente indipendente dal DOS: Windows 95. Il suo "fratello maggiore", più orientato alle reti (ed in parte ispirato ad Unix) Windows NT è disponibile dal 1994.

La parte principale del sistema MSDOS è contenuta in tre file:  
MSDOS.SYS IO.SYS COMMAND.COM

**IO.SYS** si occupa del rapporto con l'hardware e contiene i driver per le seguenti periferiche console display e tastiera stampante comunicazione seriale clock boot disk drive

**MSDOS.SYS** si occupa del rapporto tra MSDOS e le applicazioni. Gestisce quindi: organizzazione di file e record organizzazione della memoria input and output dei caratteri esecuzione dei programmi

**COMMAND.COM** interpreta i comandi inviati dall'utente, presenta il prompt, e contiene i comandi interni.

Un dischetto dove è presente il sistema MSDOS deve contenere questi tre file.  
Il resto del sistema operativo consiste in una serie di utility o comandi esterni.

- Come praticamente tutti i sistemi operativi per personal computer del periodo, l'MS-DOS era un sistema operativo monoutente e monotask
- La digitazione dei comandi consente di eseguire attività nel computer senza utilizzare l'interfaccia grafica di Windows
- Viene generalmente utilizzato solo dagli utenti più esperti.
- Nonostante oramai sembri vecchio e superato dai nuovi e potenti sistemi operativi, Dos ancora oggi può risultare utile





Molti comandi di Linux usati al prompt della shell sono simili a quelli usati in MS-DOS. Alcuni sono addirittura identici.

Funzione del Comando	MS-DOS	Linux	Esempi base di Linux
Copiare un file	copy	cp	<code>cp questofile.txt /home/questadirectory</code>
Spostare file	move	mv	<code>mv questofile.txt /home/questadirectory</code>
Elencare file	dir	ls	ls
Cancellare una schermata	cls	clear	clear
Chiudere una shell	exit	exit	exit
Visualizzare o impostare la data	date	date	date
Cancellare file	del	rm	<code>rm questofile.txt</code>
Copiare l'output sullo schermo	echo	echo	<code>echo questo messaggio</code>
Modificare un file con un editor di testi semplici	edit	gedit([a])	<code>gedit questofile.txt</code>
Paragonare il contenuto dei file	fc	diff	<code>diff file1 file2</code>
Trovare una stringa di testo in un file	find	grep	<code>grep questa parola o frase questofile.txt</code>
Formattare un dischetto	format a: (se il dischetto ◆ in A:)	mke2fs o mformat([b])	<code>/sbin/mke2fs /dev/fd0 (/dev/fd0 ◆ l'equivalente Linux di A:)</code>
Visualizzare l'help di un comando	comando /?	man([c])	<code>man comando</code>
Creare una directory	mkdir	mkdir	<code>mkdir directory</code>
Visualizzare un file	more	less([d])	<code>less questofile.txt</code>
Rinominare un file	ren	mv([e])	<code>mv questofile.txt quelfile.txt</code>

- Il termine prompt dei comandi è riferito alla parentesi angolare chiusa '>' che indica che l'interfaccia della riga di comando è pronta per accettare comandi
- Tutto ciò che c'è scritto prima del segno '>' indica la directory corrente di lavoro
  - Esempio: Se all'apertura della finestra del prompt dei comandi viene visualizzato il prompt:  
C:\documenti> significa che i comandi che immetteremo verranno eseguiti nella cartella documenti dell'unità C del computer.



- Per accedere alla linea di comando del Dos o meglio alla finestra di Windows che contiene il prompt, occorre:
  - cliccare su “Start -> Esegui” e digitare cmd
  - oppure cliccare su “Start > Programmi > Accessori > Prompt dei comandi”
- Per visualizzare un elenco di comandi comuni, dal prompt dei comandi digitare help e premere INVIO
- Prima di imparare i comandi, è importante conoscere la sintassi



## ALCUNI COMANDI UTILI:

- `cd nome_cartella`
  - Il comando `cd` permette di cambiare cartella.
- `chdir nome_cartella`
  - Il comando `chdir` cambia la directory corrente e ne visualizza il nome
- `chkdsk`
  - Il comando `chkdsk` verifica il disco e visualizza una relazione sul suo stato.
- `dir`
  - Il comando `dir` visualizza l'elenco dei file e delle sotto directory di una directory.



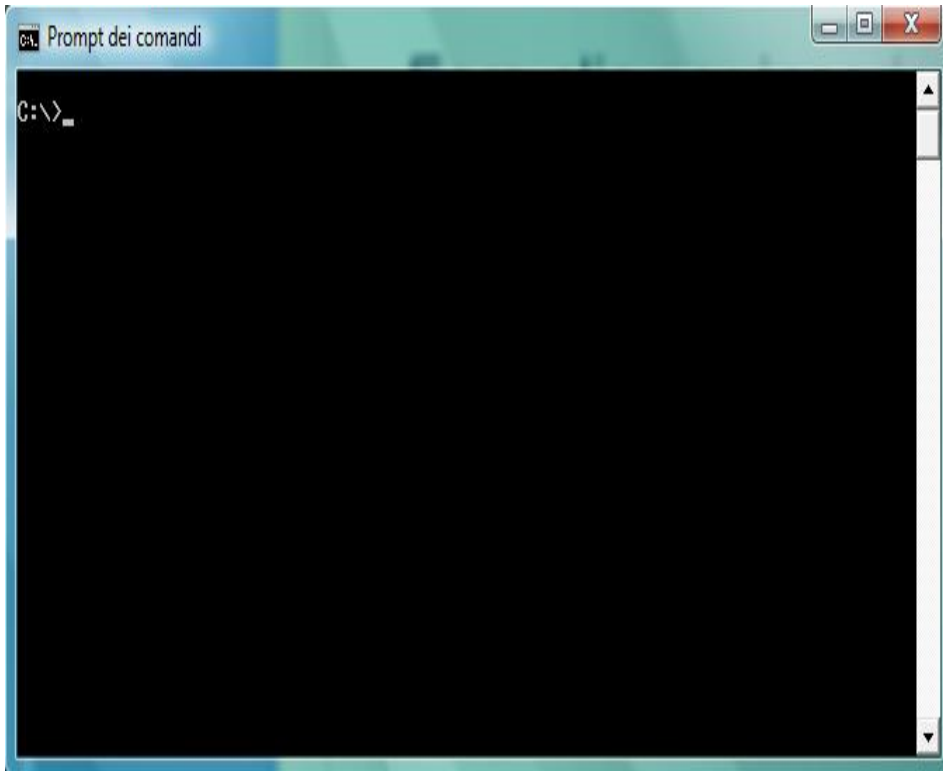
- `copy`
  - Il comando `copy` permette la copia di file o gruppi di file.
- `del nome_file`
  - Il comando `del` elimina uno o piu' file.
- `rd nome_cartella`
  - Il comando `rd` elimina uno o piu' directory.
- `dir`
  - Il comando `dir` visualizza l'elenco dei file e delle sotto directory di una directory.



- Alcuni tasti utili:
  - tasto F3: scrive sul prompt l'ultimo comando eseguito, senza però eseguirlo.
  - tasto F7: visualizza la cronologia dei comandi scritti precedentemente nel prompt. I comandi si possono riutilizzarli senza doverli scrivere.
  - tasto F8: scorre i comandi eseguiti a ritroso, partendo dal più recente. Se prima di premere F8 si digitano alcuni caratteri vengono mostrati solo i comandi che così iniziano.
  - tasto tabulazione: completamento automatico del comando



## La shell di DOS istruzioni fondamentali:



`cd\` (porta in C: )

`cd nome_cartella` “cambia directory”

`cd..` “Riporta alla percorso precedente”

`dir` “esegue il listato dei file contenuto nella cartella”

`mkdir nome_cartella` “crea una cartella nuova”

`help` “listato delle istruzioni”

## Tra i sistemi operativi per computer desktop si citano Microsoft Windows, MacOS, Linux

Windows. Nato come un'estensione di MS-DOS, anni dopo diventato un sistema operativo a tutti gli effetti, in pochi anni è riuscito a diventare uno fra i sistemi operativi più diffusi al mondo. Cosa lo rende appetibile? La miriade di programmi disponibili, la compatibilità con moltissime periferiche e anche l'ambiente facile e intuitivo. Quali sono i contro? Un primo aspetto i programmi malevoli, virus, malware ecc. Essendo il sistema operativo più diffuso al mondo è anche il sistema operativo più soggetto a questo tipo di attacchi. Questo perché? Perché nonostante sia presentato come un prodotto di eccellente qualità presenta numerose falle nella sicurezza, anche se poi vengono risolti certi problemi con aggiornamenti specifici.

Passiamo a Linux, cosa lo rende appetibile? Dal mio punto di vista il fatto che è open source, cioè non solo è gratuito ma il suo codice sorgente è disponibile a tutti perciò ognuno può modificarlo secondo le sue esigenze o può dare un contributo a renderlo migliore. L'unione fa la forza e in effetti Linux non è altro che il risultato della collaborazione di moltissimi utenti che lavorano assieme per renderlo sempre più affidabile. Un secondo aspetto è il fatto che non esistono virus per Linux, o almeno può essere attaccato ma non con la stessa facilità e con gli stessi metodi con i quali può essere attaccata una macchina Windows, inoltre se viene scoperta una falla nel sistema può essere segnalata e aggiustata. Ricordo anche che qualsiasi virus fatto per Windows non ha nessun effetto su Linux. Passiamo al software, anche questo gratuito e di ottima qualità, infatti viene usato nei server, nelle aziende, nelle scuole, nella polizia cosa si può volere di più?

Eccoci arrivati al MAC OS X. infatti Mac è un sistema affidabile, privo di virus come Linux, e veloce. I difetti supporto solo AMD e inoltre il fatto che è un sistema operativo come Windows, con il codice sorgente non disponibile a tutti.





## Richard Matthew Stallman

↓  
GNU

## Linus Torvalds

↓  
Linux

## Mark Shuttleworth

↓  
Ubuntu

"umanità verso gli altri"

«Ubuntu: Linux per gli esseri umani»

Posizione ↕	Rmax Rpeak (PFLOPS) ↕	Nome computer ↕	Modello ↕	Architettura CPU ↕	Interconnessione ↕	Produttore ↕	Site country, year ↕	Sistema operativo ↕
1 ▲	122.300 187.659	Summit	Power System AC922	POWER9, Tesla V100	Infiniband EDR	IBM	Oak Ridge National Laboratory 🇺🇸 Stati Uniti, 2018	Linux (RHEL)
2 ▼	93.015 125.436	Sunway TaihuLight	Sunway MPP	SW26010	Sunway <sup>[4]</sup>	NRCPC	National Supercomputing Center in Wuxi 🇨🇳 Cina, 2016 <sup>[4]</sup>	Linux (Raise)
3 ▲	71.610 119.194	Sierra	Power System S922LC	POWER9, Tesla V100	Infiniband EDR	IBM	Lawrence Livermore National Laboratory 🇺🇸 Stati Uniti, 2018	Linux (RHEL)
4 ▼	61.445 100.679	Tianhe-2A	TH-IVB-FEP	Xeon E5-2692 v2, Matrix- 2000	TH Express-2	NUDT	National Supercomputing Center in Guangzhou 🇨🇳 Cina, 2013	Linux (Kylin)
5 ▲	19.880 32.577	AI Bridging Cloud Infrastructure <sup>[5]</sup>	PRIMERGY CX2550 M4	Xeon Gold 6148, Tesla V100	Infiniband EDR	Fujitsu	National Institute of Advanced Industrial Science and Technology 🇯🇵 Giappone, 2018	Linux
6 ▼	19.590 25.326	Piz Daint	Cray XC50	Xeon E5-2690 v3, Tesla P100	Aries	Cray Inc.	Swiss National Supercomputing Centre 🇨🇭 Svizzera, 2016	Linux (CLE)
7 ▼	17.590 27.113	Titan	Cray XK7	Opteron 6274, Tesla K20X	Gemini	Cray Inc.	Oak Ridge National Laboratory 🇺🇸 Stati Uniti, 2012	Linux (CLE, SLES)
8 ▼	17.173 20.133	Sequoia	Blue Gene/Q	A2	Custom	IBM	Lawrence Livermore National Laboratory 🇺🇸 Stati Uniti, 2013	Linux (RHEL e CNK)
9 ▼	14.137 43.903	Trinity	Cray XC40	Xeon E5-2698 v3, Xeon Phi 7250	Aries	Cray Inc.	Los Alamos National Laboratory 🇺🇸 Stati Uniti, 2015	Linux (CLE)
10 ▼	14.015 27.881	Cori	Cray XC40	Xeon Phi 7250	Aries	Cray	National Energy Research Scientific Computing Center 🇺🇸 Stati Uniti, 2016	Linux (CLE)