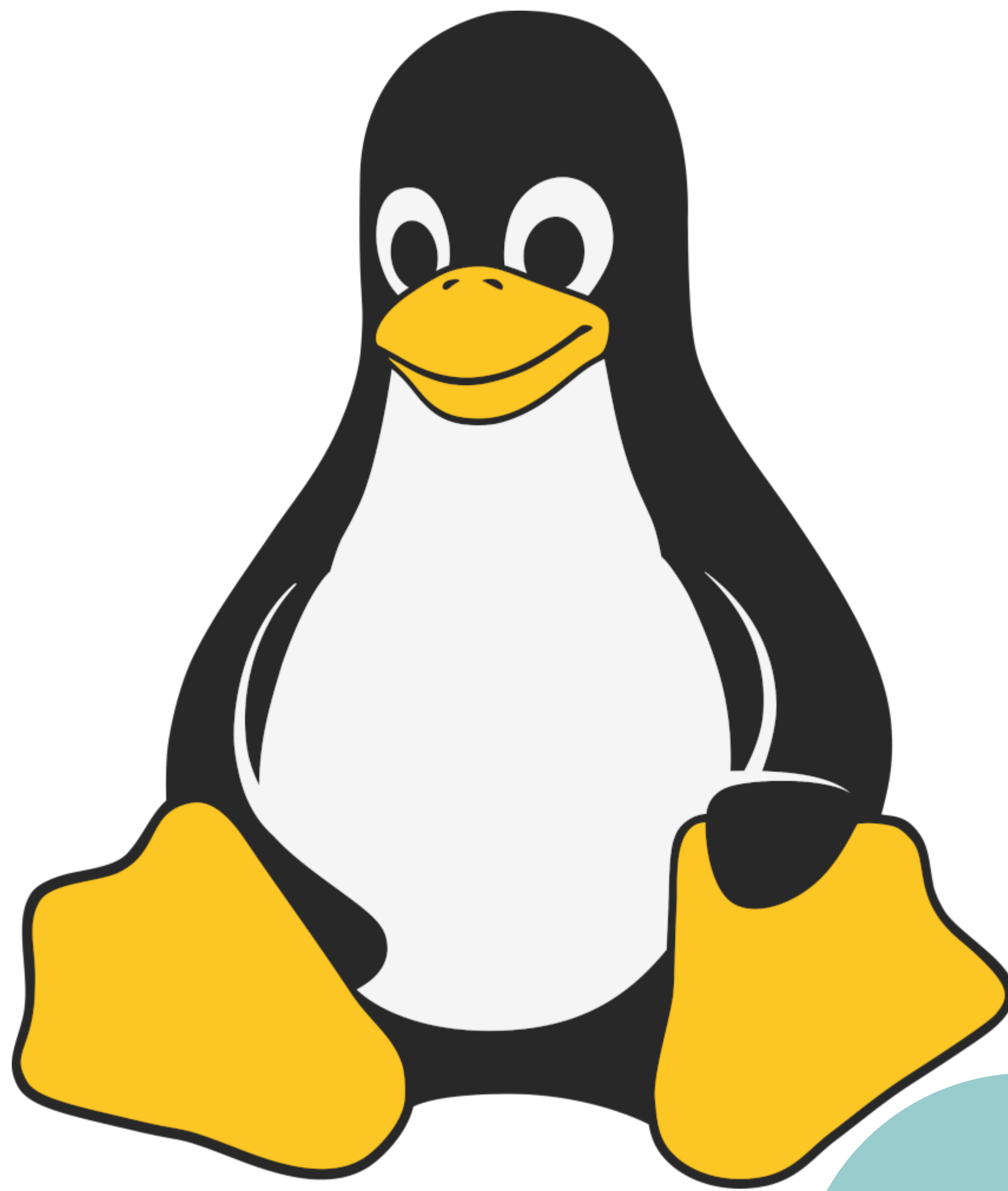


# LINUX



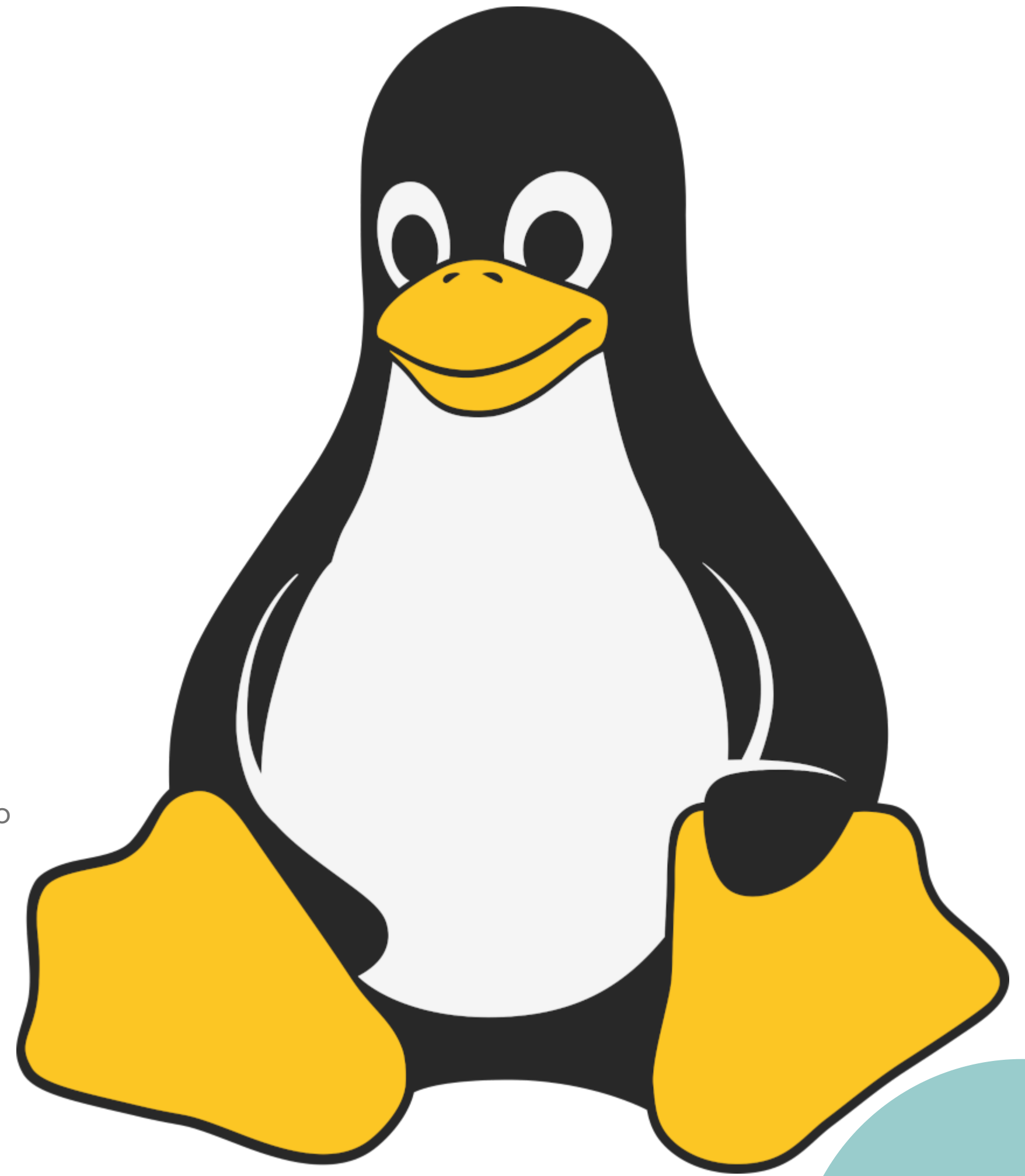
# Cos'è Linux

Linux è un sistema operativo **Unix Like**, ideato da Linus Torvalds nel 1991.

Linux è una reale alternativa al mondo Microsoft e Unix, presente in milioni di dispositivi e server, utilizzato da migliaia di sviluppatori in un mercato in continua espansione.

Linux è oggi utilizzato in sistemi integrati, per il controllo di dispositivi robotici e sullo shuttle: praticamente gira su oggetti elettronici di tutti i tipi, dai server agli smartphone, risultando il sistema operativo in assoluto più utilizzato.

Con Linux viene gestita la maggior parte di Internet, le più grandi piazze finanziarie del mondo, nonché i supercomputer che permettono le scoperte scientifiche.





# Accenni di Storia

## Linus Torvalds e la nascita di Linux

Linus Torvalds, studente finlandese di informatica, il 25 agosto del 1991, insoddisfatto del sistema operativo Minix (sistema unix-like destinato alla didattica e che mal supportava la nuova architettura i386 a 32bit), decide di creare un kernel unix con lo scopo di divertirsi e studiare il funzionamento del suo computer, un 80386.

All'inizio, il kernel programmato da Linus, chiamato Linux, aveva la necessità di girare all'interno dello userspace di Minix. Questo implicava l'adesione alla licenza del Minix stesso, ovvero il libero utilizzo esclusivamente a fini didattici, impedendone la libera distribuzione. Linus decide quindi di sostituire quella parte del sistema operativo con il software del progetto GNU. Per fare ciò, Torvalds, cambiò la licenza e adottò la GPL.

Sin dalla versione 0.01 si poteva compilare e far partire la shell GNU Bash. Linus rende pubblico il suo lavoro ed il suo codice, trovando presto altri sviluppatori che aderiscono alla visione ed al progetto, rendendo in breve tempo linux un sistema operativo completo, pienamente funzionante e libero. Il 12 marzo del 1994, Linus Torvalds presenta all'Università di Helsinki la prima versione stabile di Linux, la 1.0.

# Gnu Public License

Linus Torvalds racconta nel suo libro, che linux nasce per poter dare un'alternativa libera al predominio dei software proprietari. Adotta quindi una licenza il cui scopo principale è mantenere tale libertà.

La **GPL** sancisce che qualsiasi fornitore, al momento di distribuire file binari, deve anche rendere il codice sorgente leggibile e disponibile sotto gli stessi termini di licenza.

Essa garantisce agli utenti finali, come organizzazioni, imprese o semplici individui, di utilizzare, condividere e persino modificare il software.

Oggi arrivata alla terza versione, è progettata per essere applicata facilmente ai programmi di cui si detengono i relativi diritti.





# Ma cos'è nella pratica

Possiamo definirlo, impropriamente, un sistema operativo che, come i suoi principali competitor, Windows e Mac OSX, permette al software applicativo di dialogare con l'hardware e gestirne le risorse. Senza il sistema operativo, il software non potrebbe funzionare.

Il Sistema Operativo è composto da un certo numero di sottosistemi:

- il **Bootloader**, che è il software che gestisce il processo di avvio del computer;
- il **Kernel**, che è il cuore del sistema e si occupa di gestire le risorse presenti nel computer, rendendole disponibili alle applicazioni;
- i **Demoni**, che sono i vari servizi che ascoltano in background e si attivano all'avvio del computer o in seguito a specifici eventi;
- la **Shell**, ovvero un software che emula un terminale a caratteri che permette di controllare il computer;
- il **Server Grafico**, a cui è delegata la responsabilità di visualizzare le informazioni sul monitor;
- l'**Ambiente Desktop**, ovvero il sottosistema con cui l'utente interagisce con il sistema operativo.

# Le Distribuzioni

Una distribuzione è un confezionamento di vari pacchetti software coerenti fra loro, atto a rendere comoda e semplice, tramite procedure guidate, l'installazione di linux.

Tenendo a mente che:

- linux è solo un kernel, non un vero e proprio sistema operativo;
- le applicazioni eseguite all'interno della user-space hanno una propria versione e sono indipendenti da linux stesso;

possiamo sicuramente affermare che le distribuzioni aggiungono tutto il software necessario per avere un sistema operativo completo di software applicativi e di servizio, personalizzando le versioni degli stessi e organizzando al meglio la procedura di installazione.

# Quale distribuzione utilizzeremo

Le distribuzioni oggi disponibili sono diverse decine, tutte valide a modo loro e che si differenziano tra esse per delle peculiarità come l'ambiente di lavoro a cui sono destinate.

In questo corso utilizzeremo **Ubuntu 20.04 LTS**, una delle distribuzioni più semplici da usare che, grazie proprio alla semplicità di utilizzo, risulta essere la più diffusa.

Ubuntu nasce il 20 ottobre 2002 con l'obiettivo principale di portare linux alle masse. È una distribuzione fortemente pensata per essere efficiente ma allo stesso tempo semplice da utilizzare.

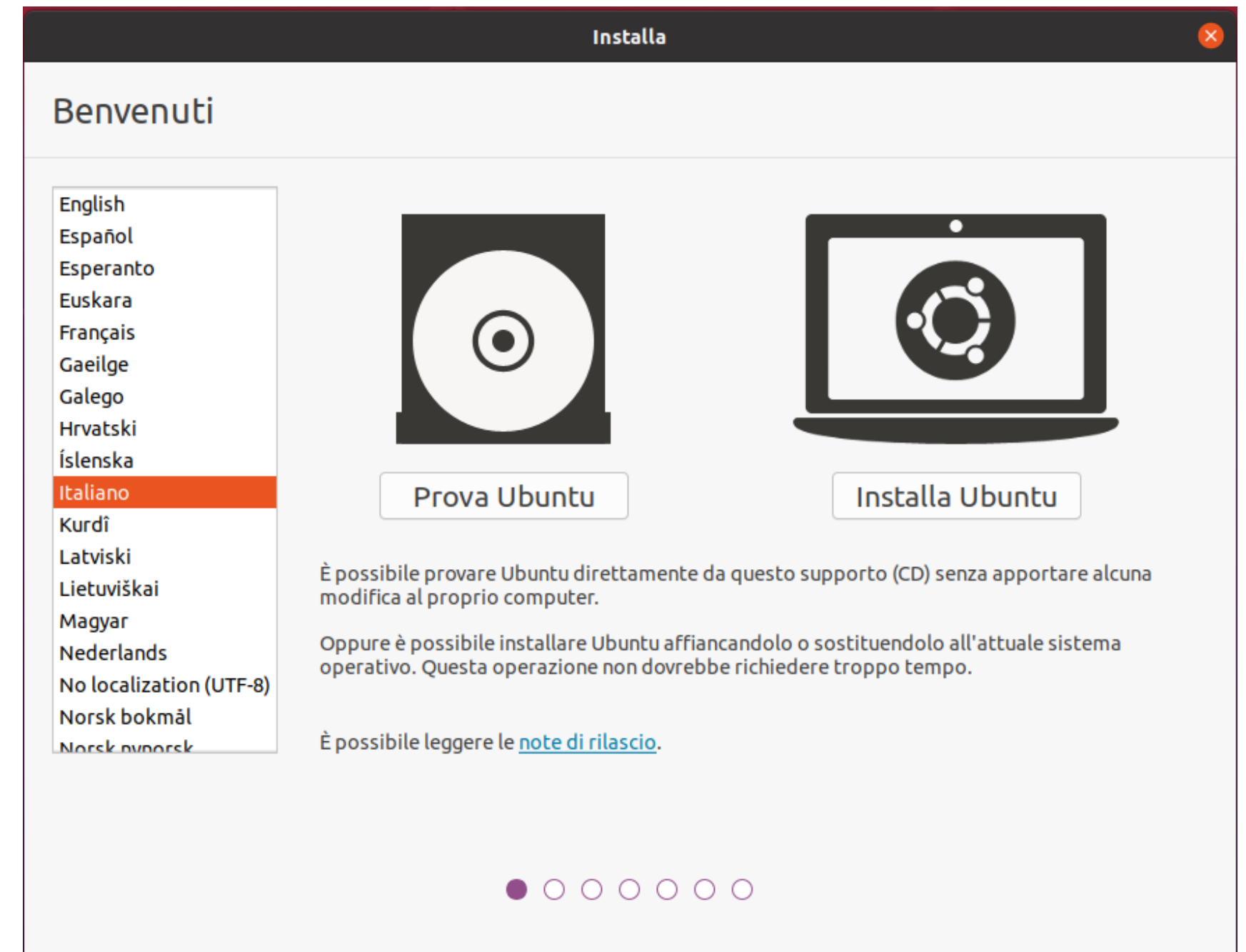
Di proprietà di Canonical, Ubuntu si presta ad essere usata sia su workstation, che su server e infrastrutture cloud.

Proprio da questi ultimi due settori, Canonical trae i propri profitti.

# Installazione

L'installazione avverrà tramite una procedura guidata, durante la quale ci verranno poste alcune domande.

**Attenzione** La procedura di installazione apporterà modifiche alla memoria del computer, riducendo lo spazio disponibile del sistema operativo già presente, se si intende mantenerlo, piuttosto che eliminando tutti i dati presenti in esso se si intende sostituirlo. È quindi necessario prestare la dovuta attenzione prima di procedere e assicurarsi di avere una copia di backup dei propri dati.





# Installazione

Installa

Aggiornamenti e altro software

Quali applicazioni installare?

☒ Installazione normale

Browser web, strumenti, software d'ufficio, giochi e riproduttori multimediali.

☐ Installazione minima

Browser web e strumenti di base.

Altre opzioni

☒ Scaricare gli aggiornamenti durante l'installazione di Ubuntu

Ciò consente di risparmiare tempo dopo l'installazione.

☐ Installa software di terze parti per grafica, dispositivi Wi-Fi e formati multimediali aggiuntivi

Questo software è soggetto alle condizioni di licenza incluse con la sua documentazione. Alcune sono di tipo proprietario.

Esci Indietro Avanti

Decidiamo se procedere con un'installazione minimale o completa e, soprattutto, se scaricare gli aggiornamenti durante l'installazione ed installare driver e software di terze parti.

Installa

Tipo d'installazione

Non è stato trovato alcun sistema operativo installato su questo computer. Come procedere?

☒ Cancella il disco e installa Ubuntu

Attenzione: in questo modo verranno eliminati programmi, documenti, foto, musica e altri file presenti in tutti i sistemi operativi.

Caratteristiche avanzate... Nessuna selezione

☐ Altro

È possibile creare o ridimensionare le partizioni oppure scegliere più partizioni per Ubuntu.

Esci Indietro Installa

Decidiamo se cancellare il contenuto del disco o ricavare un po di spazio da quello libero.

Installa

Informazioni personali

Il vostro nome: Mario

Il nome del computer: mario-Standard-PC-Q3

Il nome utilizzato per essere identificato da altri computer.

Scegliere un nome utente: mario

Scegliere una password:

Confermare la password:

☐ Accedere automaticamente

☒ Richiedere la password personale per accedere

☐ Use Active Directory

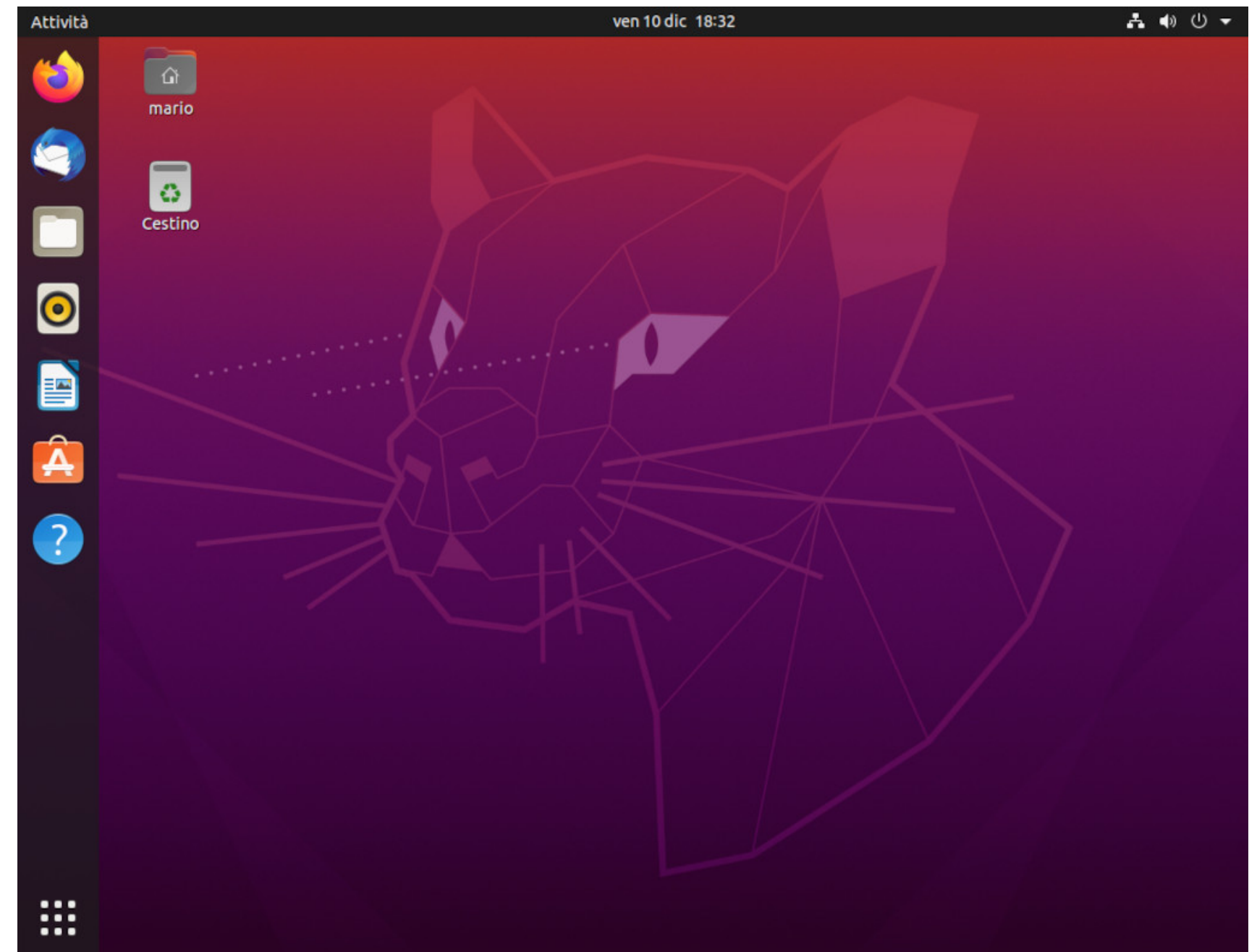
You'll enter domain and other details in the next step.

Indietro Avanti

Inseriamo il nome utente che desideriamo utilizzare, il nome con cui il computer sarà riconosciuto all'interno della rete e la password

# Primo Avvio

Terminata l'installazione, ci troveremo dinanzi una schermata di accesso con evidenziato il nome utente che abbiamo scelto nella fase precedente. Cliccandoci sopra, ci verrà richiesta la password (anch'essa scelta nella fase precedente) e ci troveremo nell'ambiente desktop predefinito di Ubuntu



# Il File System

Il file system è un sistema di archiviazione su un supporto di memoria che struttura e organizza in modo specifico scrittura, ricerca, lettura, memorizzazione, modifica ed eliminazione dei file. La strutturazione dei file deve garantire una loro identificazione rapida senza errori e un accesso il più veloce possibile da parte degli utenti. Inoltre, il file system definisce, ad esempio, le seguenti proprietà:

- convenzioni sui nomi di file
- attributi dei file
- controllo/i di accesso

Inoltre il file system è un'importante componente operativa che funge da interfaccia tra il sistema operativo e tutte le unità collegate interne ed esterne, ad esempio tramite una porta USB.

# Il File System su Linux

L'organizzazione del file system di Linux è gerarchica: tutto inizia da una singola directory, non esiste distinzione tra hardware e software (le periferiche hardware in Linux vengono rappresentate come dei file speciali) e qualsiasi parte del sistema è figlia della directory radice. In parole povere, tutto inizia dalla directory root e tutto termina con essa.

La directory `/` è accessibile digitando `cd /` da terminale, oppure scrivendo `/` nella barra degli indirizzi del file manager.

- / (directory root)
- |\_ /boot
- |\_ /dev
- |\_ /etc
- |\_ /home
- |\_ /lib
- |\_ /lib64
- |\_ /mnt
- |\_ /opt
- |\_ /proc
- |\_ /root
- |\_ /run
- |\_ /sys
- |\_ /tmp
- |\_ /usr
- |\_ /var

# Il File System su Linux

La directory `/boot` contiene tutti i file relativi all'avvio del sistema.

La directory `/dev` contiene dei file di tipo particolare che rappresentano i `devices` presenti e riconosciuti nella macchina;

La directory `/etc` contiene tutti i file di configurazione del sistema;

La directory `/home` contiene tutte le "home directory" degli utenti;

Le directory `/lib` e `/lib64` contengono le librerie di sistema, rispettivamente a 32 ed a 64bit;

La directory `/mnt` è vuota e rimane a disposizione dell'utente per montarvi manualmente i devices che vengono collegati alla macchina;

La directory `/opt` contiene tutte le applicazioni installate dall'utente che risultano essere opzionali, non facenti parte del sistema e spesso non gestiti dal package manager utilizzato dal sistema;



# Il File System su Linux

La directory `/proc` è una directory particolare: ogni processo in esecuzione in Linux, occupa delle risorse di sistema. Per gestire tali processi, il sistema utilizza una sorta di file system che non prende spazio sulla memoria di massa ma è totalmente caricato in RAM. Questo file system prende posto all'interno della directory `/proc`.

Navigando all'interno della cartella, si scopriranno tante directory quanti sono i processi in esecuzione sul sistema (alcuni nomi corrisponderanno ai nomi dei processi, altri al pid assegnato per l'esecuzione);

La directory `/root` è la home directory dell'utente root;

La directory `/run` è utilizzata dalle applicazioni per salvare tutto ciò che gli occorre per poter operare, come le informazioni sui socket, i file di lock e altre informazioni utili solamente a runtime;

La directory `/sys` è un file system virtuale dove il kernel esporta le proprie informazioni per renderle disponibili ad altri programmi;

La directory `/tmp` è dove tutti i programmi possono scrivere i file temporanei;

# Il File System su Linux

La directory `/usr` contiene file relativi alle applicazioni e al kernel

- in `/usr/bin` in genere sono presenti i file binari utilizzabili dai normali utenti;
- in `/usr/lib` sono presenti le librerie per il funzionamento dei programmi installati;
- in `/usr/include` sono presenti gli header delle librerie, utili per la compilazione dei programmi;
- in `/usr/share` sono presenti i dati relativi alle varie applicazioni, tra cui le pagine dei manuali;
- in `/usr/sbin` in genere sono presenti i file binari utilizzabili solo da root;
- in `/usr/local` spesso troviamo una struttura del tutto simile a `/usr`, al cui interno trovano posto le applicazioni compilate manualmente dall'utente finale.

Tutti gli eseguibili presenti in `/usr/bin`, `/usr/local/bin` o `/usr/sbin`, sono eseguibili da terminale senza specificarne il percorso.

La directory `/usr`, e tutte le sue sottodirectory, sono accessibili da tutti, ma modificabili soltanto dall'utente root.

La directory `/var` contiene i dati che i programmi cambiano durante la normale esecuzione del sistema.

# Il File System su Linux

Tramite il comando **stat /**, otteniamo:

```
` File: /  
Dim.: 4096 Blocchi: 8 Blocco di IO: 4096 directory  
Device: fc05h/64517d Inode: 2 Coll.: 20  
Accesso: (0755/drwxr-xr-x) Uid: ( 0/ root) Gid: ( 0/ root)  
`
```

Le prime due righe indicano che la directory si chiama /, è grande 4096 bytes (come ogni directory) e occupa 8 blocchi.

Nella terza troviamo il device in cui risiede e il numero di ulteriori directory che contiene.

Nell'ultima riga sono presenti i permessi, l'utente e il gruppo a cui appartiene.

# Il File System su Linux

I permessi associati alla root sono **drwxr-xr-x**, dove:

- la **d** sta per directory;
- il primo gruppo di tre caratteri **rw****x** indica i permessi di cui gode il proprietario;
- il secondo gruppo, **r****-x**, indica i permessi di cui gode il gruppo;
- l'ultimo gruppo, **r****-x**, i permessi di cui gode qualunque altro utente.

I permessi **r**, **w** e **x**, indicano rispettivamente:

**r** - diritto di lettura

**w** - diritto di scrittura

**x** - diritto di esecuzione

Il proprietario di `/` è l'utente con identificativo 0, quindi root, e il gruppo proprietario è il gruppo 0, ovvero il gruppo root.

# File System – Links

Le informazioni riguardanti un oggetto (file o directory) di un filesystem, sono contenute in un inode, che viene identificato da un numero progressivo e descrive le caratteristiche base di un oggetto: permessi, data di modifica, tipo, posizione, ecc.

Il comando `ln sorgente destinazione` crea un link (alias) di un file o di una directory

Quando si opera sul link è come se si operasse direttamente sul file, tranne quando si rimuove il link, ovvero non si elimina il file.

I link possono essere di due tipi: **hard** e **soft**:

- gli hard link sono di fatto un altro nome per un file esistente; originale e link sono indistinguibili, condividono lo stesso inode e non possono risiedere su due file system diversi o far riferimento a directory;
- i soft link, creabili con l'opzione `-s`, sono un tipo speciale di file che fanno riferimento a un inode diverso, può essere fatto su file system diversi e su file che non esistono;

L'uso di link è completamente trasparente e non impatta sulle performance del sistema.



# File System - Gestione

Prima di poter utilizzare un filesystem (ad es. CDROM, USB, condivisione di rete windows, directory nfs), questo deve essere formattato e montato in una sotto-directory della root ( / ).

Una volta montato, il file system risulta accessibile a programmi ed utenti in maniera totalmente trasparente, diventando parte integrante dell'albero delle directory sotto /.

Dopo l'uso, il file system può essere smontato, operazione necessaria per le unità di memorizzazione removibili.

La directory in cui viene montato un file system può anche non essere vuota ma, il suo contenuto non potrà essere visibile sino a quando non si smonta il file system

# File System – Gestione – Comandi

``mount -t [tipo fs] [opzioni] device destinazione`` Monta un dispositivo a blocchi nella directory di destinazione

``umount [opzioni] device`` Smonta un dispositivo

``df [opzioni] file`` Verifica lo spazio libero su disco

``du [opzioni] file`` Verifica lo spazio occupato da file e directory

``fsck [opzioni] device`` Verifica l'integrità e ripara il file system

``mkfs [opzioni] device`` Crea un file system, ovvero lo formatta

Nel file ``/etc/fstab`` vengono configurate le informazioni sui vari file system preimpostati sul sistema, impostandone i punti di mount, il tipo e altri utili parametri

# File System – Gestione – Comandi

``chown [opzioni] [utente] file`` Assegna un nuovo utente proprietario, sia specificandone il nome piuttosto che l'UID

``chgrp [opzioni] [gruppo] file`` Assegna un nuovo gruppo proprietario, similmente a quanto avviene per il comando precedente

``chmod [opzioni] [modo] file`` Modifica i permessi sul file, sia usando l'octal mode, dove si usano tre numeri, riferiti rispettivamente a owner, group e others, sia utilizzando il symbolic mode, dove si usano le lettere r (read), w (write), x (execute) per assegnare i permessi a u (user), g (group) e o (others).

I seguenti comandi sono equivalenti:

```
`  
chmod 755 nomefile  
chmod u=rwx,g=rx nomefile`
```

# File System – Copia/Modifica file e directory

``cp [opzioni] sorgente destinazione`` Copia di file o directory

``mv [opzioni] sorgente destinazione`` Spostamento di file o directory

``rm [opzioni] file`` Rimozione di file o directory

``ls [opzioni] path`` Elenca contenuto di una directory

``tree [opzioni] path`` Elenca contenuto di una directory con visualizzazione ad albero

``cd path`` Cambia directory corrente

``mkdir [opzioni] directory`` Crea una directory

``rmdir [opzioni] directory`` Elimina una directory

# Multiutenza e Permessi

Linux è un sistema operativo multiutente, dove differenti utenti possono avere accesso al sistema in modo contemporaneo, avendo i propri dati, i propri programmi e impostazioni.

L'operazione di autenticazione dell'utente avviene tramite nome utente e password e il login può essere eseguito in ambienti diversi, grafici o testuali, da locale o da remoto

L'utente principale del sistema è l'utente **root**, amministratore del sistema, con tutti i poteri che comporta questo ruolo. Ha poteri assoluti sul sistema:

- Aggiungere, eliminare e modificare account
- Installare e configurare servizi
- Accesso completo a tutti i file presenti nel file system
- Aggiungere o modificare file system
- Distruggere tutto con un solo comando (il potere di root sul sistema è assoluto)



# Multiutenza e Permessi

Buona parte delle distribuzioni moderne, hanno deciso di disattivare in modo predefinito l'account di amministrazione, incoraggiando l'utente ad utilizzare lo strumento `sudo` per svolgere i compiti di amministrazione di sistema.

Lo strumento `sudo` permette a un utente (autorizzato) di elevare temporaneamente i propri privilegi usando la propria password. Per poterlo fare, l'utente deve appartenere al gruppo admin ed essere aggiunto al file `/etc/sudoers`.

Se per qualche motivo si rende necessario abilitare l'account di root, basterà assegnargli semplicemente una password con il comando **sudo passwd**

In modo predefinito, l'utente iniziale creato durante l'installazione di Ubuntu è un membro del gruppo `admin` ed è stato aggiunto al file `/etc/sudoers` come utente autorizzato all'utilizzo di sudo.

# Shell

Per shell si intende l'interfaccia testuale tramite la quale l'utente può operare sul sistema. Essa si occupa di gestire la comunicazione fra utente e sistema operativo, interpretando ed eseguendo i comandi dell'utente.

Può avere diversi utilizzi:

- **Uso interattivo**, nel quale il sistema attende i comandi digitati dall'utente;
- **Configurazione della sessione**, con cui vengono definiti variabili e parametri utilizzati in ogni interazione dell'utente con la macchina;
- **Programmazione**, con la quale, utilizzando comandi di sistema e funzionalità della shell, è possibile realizzare piccoli programmi (script shell), in grado di automatizzare operazioni e reagire ad eventi.

# Bash

Bash (acronimo per Bourne Again Shell) è una shell del progetto GNU usata nei sistemi operativi Unix e specialmente in GNU/Linux.

Si tratta di un interprete di comandi che permette all'utente di comunicare con il sistema operativo attraverso una serie di funzioni predefinite o di eseguire programmi.

Tecnicamente bash è un clone evoluto della shell standard di Unix (`/bin/sh`), chiamata anche Bourne Shell dal nome del suo autore originario, Stephen Bourne.

Una delle funzionalità maggiormente apprezzate di bash è quella della redirectione dell'input e dell'output, grazie al quale è possibile eseguire più programmi in cascata, passando come input dell'uno l'output dell'altro.

# Bash – Programmazione

## Variabili

Le variabili in bash scripting non hanno bisogno di essere inizializzate, l'assegnazione avviene utilizzando la sintassi `VAR=valore`.

È molto importante che il valore sia scritto subito dopo l'uguale senza alcuno spazio.

Il riferimento a tutte le variabili si fanno antepponendo il caratter `$` al nome della variabile: `VAR=world`  
`echo hello $VAR!`

# Bash – Programmazione

## Variabili Speciali

``$N`` dove posto N è un numero intero, corrisponde dell'argomento N-esimo passato da terminale al programma (`$0` è il nome del programma).

``$*`` stringa che contiene tutti gli argomenti passati da terminale al programma.

``$#`` stringa che contiene tutti il numero degli argomenti passati da terminale al programma.

``$@`` stringa che contiene tutti gli argomenti passati allo script corrente (ogni paramentro viene opportunamente quotato, questo permette l'utilizzo di questa vairabile nei cicli for).

``$?`` stringa che contiene il valore di uscita dell'ultimo comando o funzione. Il comando ha successo se ritorna zero, qualsiasi altro valore indica invece un codice di errore.

``$$``, stringa che contiene il PID del processo in esecuzione.



# Bash – Script

## Creazione di un semplice file di script bash

Per iniziare sarà necessario creare un nuovo file vuoto, in cui andremo ad inserire le istruzioni che bash dovrà eseguire.

Il comando per creare un file vuoto su linux è `touch nomefile.sh`, dove *nomefile* è il nome che vorremo assegnargli.

Creato il file, utilizziamo un editor di testo sufficientemente semplice: **nano**

Digitiamo quindi `nano nomefile.sh` e iniziamo a scrivere il seguente testo:

```
~  
#!/bin/bash  
touch ciao_linux.txt  
echo "Imparo linux!" > ciao_linux.txt  
~
```

e quindi salviamo e chiudiamo il file con la sequenza di comandi CTRL+O e CTRL+X

A screenshot of the GNU nano 5.8 text editor interface. The title bar at the top shows 'GNU nano 5.8', the filename 'nomefile.sh', and the status 'Modificato'. The editor content shows a bash script with the following lines: `#!/bin/bash`, `touch ciao_linux.txt`, and `echo "Imparo linux un pinguino alla volta!" > ciao_linux.txt`. The bottom status bar displays various keyboard shortcuts: `^G` Help, `^O` Salva, `^W` Cerca, `^K` Cut, `^T` Execute, `^C` Location, `^X` Esci, `^R` Inserisci, `^V` Sostituisci, `^U` Paste, `^J` Giustifica, and `^_` Vai a riga.

# Bash – Script

## Creazione di un semplice file di script bash

Nella prima riga indichiamo al sistema qual'è il programma con il quale vogliamo interpretare il contenuto dei comandi.

In questo caso si tratta appunto di bash.

Nella seconda diamo il comando per creare un file vuoto con il nome ``ciao_linux.txt``

Nella terza diamo il comando ``echo``, la cui funzione è quella di ridare in output la stringa che riceve in input. L'output di echo lo reindirizziamo verso il file appena creato.

```
GNU nano 5.8      nomefile.sh      Modificato
#!/bin/bash

touch ciao_linux.txt
echo "Imparo linux un pinguino alla volta!" > ciao_linux.txt
```

^G Help ^O Salva ^W Cerca ^K Cut ^T Execute ^C Location  
^X Esci ^R Inserisci ^\ Sostituisci ^U Paste ^J Giustifica ^\_ Vai a riga

# Bash – Script

## Creazione di un semplice file di script bash

Creato il file, non ci resta che assegnargli i diritti corretti ed eseguirlo.

Per prima cosa impostiamo i diritti di esecuzione:

```
`chmod u+x nomefile.sh`
```

E quindi lanciamo lo script digitando:

```
`./nomefile.sh`
```

Eseguito il comando, sembrerà non accadere nulla. In realtà bash ha creato per noi un file denominato `ciao_linux.txt` con all'interno il testo che abbiamo dato in input ad echo.

Ci basterà aprirlo con l'editor di testo che più ci aggrada (anche nano va benissimo), oppure utilizzare un altro comando standard della shell che ci permette di stampare a video il contenuto di qualunque file:

```
`cat ciao_linux.txt`
```



```
GNU nano 5.8      nomefile.sh      Modificato
#!/bin/bash

touch ciao_linux.txt
echo "Imparo linux un pinguino alla volta!" > ciao_linux.txt

^G Help      ^O Salva     ^W Cerca     ^K Cut       ^T Execute   ^C Location
^X Esci      ^R Inserisci ^\ Sostituisci ^U Paste     ^J Giustifica ^_ Vai a riga
```

# Bash – Programmi Utili

``expr`` l'interprete bash non è in grado di eseguire operazioni matematiche, quindi questo comando ci viene in aiuto

``grep`` utile per mostrare solo alcune righe ricevute in input sulla base del parametro che farà da filtro

``cut`` serve a spezzare una stringa in tante sotto-stringhe in base a un determinato separatore

``cat`` serve per copiare, unire e manipolare il contenuto di uno o più file

``touch`` utile per creare un file vuoto

``echo`` replica in output ciò che riceve in input

``man`` mostra il manuale del comando passato come parametro

## Bash – Esempi di ``expr``

```
`  
$ var1='10'  
$ var2='20'`
```

Mostra 0 se due numeri sono uguali con operatore '='

```
` $ expr $var1 = $var2  
0`
```

Mostra 1 quando arg1 è minore di arg2

```
` $ expr $var1 \< $var2  
1`
```

Mostra 1 quando arg1 è diverso da arg2

```
` $ expr $var1 \!= $var2  
1`
```

## Bash – Esempi di `grep`

Trova tutte le linee in cui viene trovato il pattern 'volpe'

```
` $ grep "volpe" pinocchio.txt `
```

Cerca ricorsivamente il pattern "gatto" in tutti i file di una determinata cartella e in tutte le sottocartelle

```
` $ grep -r "gatto" /home/utente/libri/ `
```

Il pattern da ricercare può anche essere parziale.

```
` $ grep -r "g..to" /home/utente/libri/ `
```

## Bash – Esempi di `cut`

Estrae il primo ed il sesto campo dal file `/etc/passwd`, che utilizza il carattere `:` come separatore di campo

```
` $ cut -f 1,6 -d : /etc/passwd `
```

Estrae i primi dieci caratteri da ogni linea che giunge dallo standard input

```
` $ cut -c -10 pinocchio.txt `
```

Estrae i byte che vanno dal quinto al decimo, il sedicesimo e dal ventesimo in poi di ogni linea del file `prova.txt`

```
` $ cut -b 5-10,16,20- prova.txt `
```



# Bash – Esempi di `cat`

Mostra il contenuto del file

```
` $ cat /etc/passwd `
```

Mostra il contenuto di più file

```
` $ cat prova_1.txt prova_2.txt `
```

Crea un file con il comando cat

```
` $ cat > prova.txt
```

digito del testo

da inserire nel mio file

e lo salvo

CTRL+D `

## Bash – Esempi di `cat`

Copia il contenuto di un file in un altro

```
` $ cat prova_1.txt > prova_2.txt `
```

Accoda il contenuto di un file in un altro

```
` $ cat prova_1.txt >> prova_2.txt `
```

Unisce il contenuto di più file in uno

```
` $ cat prova_1.txt prova_2.txt > prova_3.txt `
```

Ordina le righe dell'insieme dei file elencati copiandole nel file di uscita

```
` $ cat prova_1.txt prova_2.txt prova_3.txt | sort > prova_sorted.txt `
```

# Bash – Pipes



Dati due comandi A e B, il pipe viene utilizzato per redirigere l'output del comando all'input del comando B

Il pipe viene spesso utilizzato su più comandi in linea e il suo simbolo è `|`, che viene posto tra due comandi

Mostra tutti i file contenuti nella directory corrente che contengono '.txt'

```
` $ ls -l | grep .txt `
```

Mostra il numero di processi il cui owner è root

```
` $ ps -ef | grep root | wc -l `
```

# Bash – Manipolazione delle stringhe

In Bash una stringa non è altro che una variabile, si indica dunque con `${nome}` oppure con la forma abbreviata `$nome`.

Il nome dentro le graffe può essere seguito da un modificatore che manipola la variabile, ad esempio:

```
` VAR="stringa-di-esempio" echo ${VAR#stringa-} ` elimina "stringa-" dall'inizio della variabile
```

I modificatori sono molti, questi sono i tre fondamentali:

``#`` sottrae dall'inizio della stringa

``%`` sottrae dalla fine della stringa

``/`` sostituisce una sottostringa con un'altra

# Cron

Spesso capita di dover far eseguire al sistema dei comandi ad un dato momento del giorno, della settimana o dell'anno (ad es. i processi di backup automatici).

Per fare ciò, su linux, esiste **Cron**, sotto-sistema a cui viene delegata la schedulazione delle operazione. Per fare questo, andremo ad utilizzare **crontab**.

Crontab ci permette di schedulare delle operazioni sul nostro sistema con una certa semplicità, mettendo a disposizione del demone cron le informazione per eseguire le operazioni impostate.

Ogni utente del sistema può avere la propria versione di crontab.

# Il comando crontab

La sintassi del comando è `crontab -u username -opzione`, dove con `-u` specifichiamo il nome utente che possiede il file crontab. Se l'opzione `-u` viene omessa, il sistema presumirà che si stia operando sul proprio crontab personale.

Altre opzioni sono:

- `-l` che ci permette di visualizzare il contenuto puntuale del crontab

- `-r` che rimuove tutte le operazioni eventualmente impostate

- `-e` che ci permette di editare le operazioni pianificate del nostro crontab tramite l'editor di testo predefinito

Ogni riga del file crontab definisce un lavoro da eseguire e sarà simile al seguente:

```
30 23 * * * tar -cvzf /backup_sicuro.tgz /home
```

dove i primi cinque parametri definiscono rispettivamente minuto, ora, giorno del mese, mese e giorno della settimana e il sesto il comando da eseguire.

L'asterisco significa "tutti".

# Il comando crontab

È possibile sostituire i primi cinque valori con delle stringhe "speciali" che assolvono al medesimo compito:

- `@reboot` lancia il comando all'avvio del sistema
- `@yearly` lancia il comando una volta all'anno
- `@monthly` lancia il comando una volta al mese
- `@weekly` lancia il comando una volta alla settimana
- `@daily` lancia il comando una volta al giorno
- `@hourly` lancia il comando una volta ogni ora



# Networking

Sin dalle prime versioni, per ovvi motivi, una delle funzionalità principali di linux, è quella di poter comunicare con l'esterno.

Per poterlo fare, il kernel di linux supporta svariati protocolli di comunicazione, dai più obsoleti ai più innovativi.

Sicuramente il più utilizzato è il protocollo TCP/IP.

Per protocollo TCP/IP ci si riferisce a un importante protocollo internet che è responsabile della trasmissione o del trasferimento di pacchetti di dati in rete e su internet.

Tramite esso, due computer riescono a scambiarsi informazioni sia se sono collegati alla stessa rete locale (Ethernet o WiFi), sia che si trovano ai due capi di una connessione Internet (broadband).

# Networking

Ad ogni computer collegato a una rete tramite il protocollo TCP/IP, viene assegnato un **indirizzo IP**. Questo indirizzo, nella sua versione 4, la più diffusa, ha un formato composto da quattro numeri, che vanno da 0 a 255: ``192.168.1.55``

Per verificare quale indirizzo è stato assegnato alla nostra macchina con linux, abbiamo a disposizione il comando ``ip``

# Networking

Digitando da terminale il comando ``ip a`` avremo un risultato simile al seguente ```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
default qlen 1000
link/ether 52:54:00:7c:00:3a brd ff:ff:ff:ff:ff:ff
inet 192.168.1.103/24 brd 192.168.1.255 scope global dynamic noprefixroute enp1s0
valid_lft 3445sec preferred_lft 3445sec
inet6 fe80::813a:204:8b17:1a56/64 scope link noprefixroute
valid_lft forever preferred_lft forever
`
```

# Networking

Tra le varie informazioni riportate dal comando ip, vi sono:

- il nome dell'interfaccia di rete, in questo caso **lo** e **enpls0**;
- lo stato dell'interfaccia, in questo caso **UNKNOWN** per lo e **UP** per enpls0;
- l'indirizzo di rete assegnato, **127.0.0.1/8** per lo e **192.168.1.103/24** per enpls0;

# Networking – lo

L'interfaccia di **loopback** o **lo**, è un'interfaccia virtuale e fittizia, necessaria al normale funzionamento dell'ambiente linux.

Tramite questa interfaccia, i vari servizi che fanno uso del protocollo TCP/IP, possono dialogare tra loro e scambiarsi informazioni in tutta sicurezza.

L'indirizzo assegnato a questa interfaccia è standard e prefissato: **127.0.0.1/8**

## Networking – enp1s0, eth0, altro

Ogni altra interfaccia, su linux, può avere o un nome generico (come ad es. eth0, eth1, ect.) oramai in disuso, o un nome specifico relativo all'hardware utilizzato (come ad esempio enp1s0).

Tramite questa o altre interfacce, il computer può comunicare correttamente con l'esterno, usufruire dei servizi messi a disposizione dagli altri sistemi, e metterne a sua volta a disposizione.

Non esiste un indirizzo prefissato da assegnargli, esso dipende dalla rete a cui si è collegati e quindi dalle politiche messe in atto dall'amministratore di rete o, più semplicemente, dal router tramite il quale guadagniamo la connessione alla rete.

# Networking – file di configurazione

## **/etc/sysconfig/network**

Contiene le principali configurazioni come il nome della macchina (hostname) e il default gateway. Ad esempio: ~

```
NETWORKING=yes ;Attiva la rete
```

```
HOSTNAME=ubuntu-pc ;Il nome della macchina
```

```
GATEWAY=192.168.1.1 ;L'indirizzo IP del router che ci permette la connessione a internet
```

```
~
```

## **/etc/sysconfig/network-script/\***

Directory contenente i file di configurazione delle singole interfacce

## **/etc/hosts**

Contiene il mapping tra indirizzi e hostname (ed alias). Ad esempio: ~

```
127.0.0.1 localhost.localdomain localhost
```

```
192.168.1.55 ubuntu-pc
```

```
~
```



# Networking – file di configurazione

## **/etc/services**

Contiene il mapping tra il numero delle porte e i nomi dei servizi.

È un file che solitamente non si modifica, a meno di creare nuovi servizi personalizzati.

## **/etc/resolv.conf**

File di configurazione del client DNS, ovvero contiene gli indirizzi del server DNS e un possibile dominio dell'host e l'ordine di ricerca. Ad esempio: ~

```
nameserver 192.168.1.1 ;IP del DNS primario, che spesso coincide con l'indirizzo del  
default gateway;
```

```
nameserver 8.8.8.8 ;IP del DNS secondario
```

```
domain miodominio.it ;Nome del dominio dove si trova l'host
```

```
options timeout:6 ;Imposta il timeout a 6 secondi per le query DNS ~
```

# In definitiva, perchè usare linux

Linux è un sistema operativo libero, non ha costi di licenza e c'è una vera e propria comunità di sviluppatori che giornalmente si adopera per migliorarlo, aggiornarlo e diffonderlo.

È semplice da utilizzare, la sua interfaccia grafica è molto simile a qualunque altro sistema operativo: cartelle, copia e incolla, browser sono uguali ai programmi presenti negli altri sistemi operativi

La semplicità però non ci impedisce di addentrarci nella complessità dei sistemi server, nell'esplorare funzionalità che tutti i giorni siamo abituati ad utilizzare come semplici utenti.

Inoltre linux, ad oggi, è esente da virus informatici: è di fatto l'unico sistema operativo struttura in modo "sicuro" e che ci permette un'immunità dai virus molto vicina al 100%.

