Appunti di LAS

Sebastiano Filippetto

Finito di scrivere in data 19 maggio 2017
Per eventuali suggerimenti e correzioni: seba.fil@live.com

Indice

Capitolo 1

Introduzione

Sistema: Insieme di elementi che interagiscono tra loro in modo coordinato per poter raggiungere i risultati per i quali il sistema stesso è stato definito.
Un sistema che funziona non si cambia, non si tocca, non si modifica: si mantiene.
Un sistema funziona quando fa quello per cui è progettato, in tempi "umani".
Un sistema si cambia solo quando non fa più quello che deve fare oppure il nuovo sistema ha dei vantaggi palesemente superiori rispetto al vecchio.

Cosa fa un Sys Admin?

- Deve conoscere la componentistica hardware
- Deve conoscere il software
- Deve conoscere gli strumenti del sistema operativo
- Deve gestire gli utenti (o utonti)
- Deve occuparsi degli accessi al sistema
- Deve gestire i vari tecnici che fanno manutenzione agli apparati
- Deve avere fin troppa pazienza
- Deve occuparsi della sicurezza dei dati
- Deve pianificare delle efficaci strategie di backup e restore
- Deve sorvegliare il sistema
- Deve convincere gli utenti che i computer non sono artefatti esoterici e che non si viene pagati per fare da psicologo ai colleghi.

Perché gli utenti sono i nemici del Sys Admin? Beh, molto semplicemente perché i loro PC vengono colpiti in qualsiasi maniera da qualsiasi tipo di incantesimo maligno, ma non sembrano ricordarsi quale al momento della segnalazione.
È giusto imparare che le responsabilità degli utenti sono le seguenti:

- l'uso e l'abuso delle risorse
- la condivisione e la segretezza dei dati personali, in particolare delle informazioni di autenticazione
- l'integrità e la disponibilità dei dati personali, in particolare il regolare svolgimento dei backup
- la rivelazione non autorizzata di informazioni riservate
- lo scambio di posta elettronica con forum o gruppi di discussione controversi
- Le regole di buona condotta nell'uso della posta elettronica

Ma come sappiamo tutti, da grandi responsabilità derivano... No aspetta, non era così... Pazienza, l'utente ha (purtroppo) dei diritti:

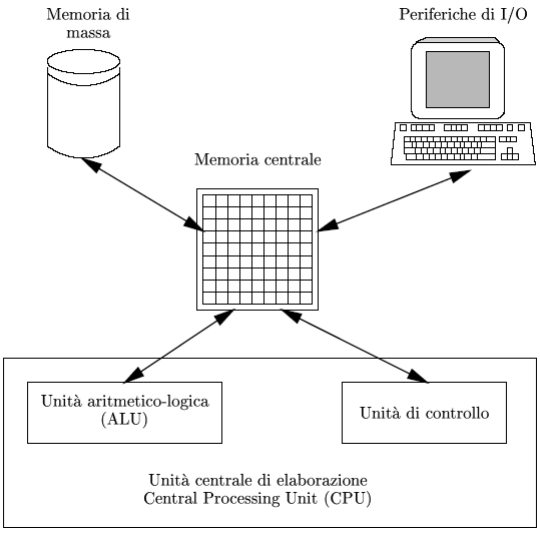
- La riservatezza dei dati personali
- La riservatezza della posta elettronica
- Lavorare in un ambiente funzionale e funzionante
- Fare domande lecite
- Avere assistenza
- Lamentarsi

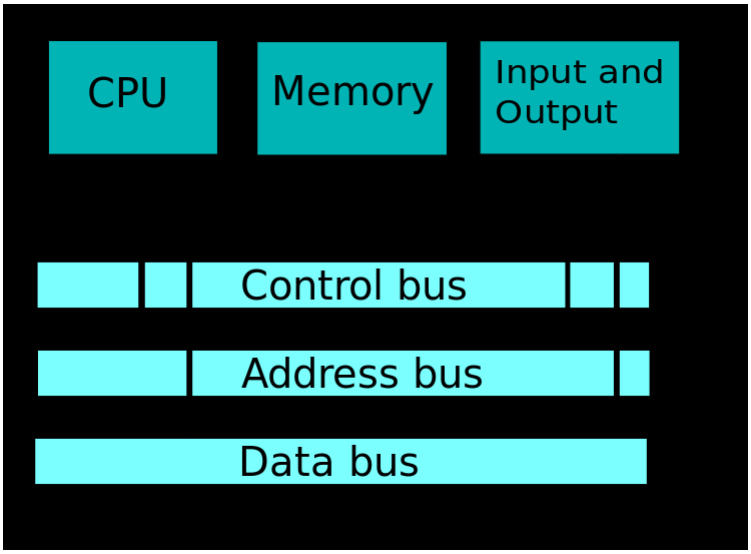
Cosa deve saper fare, in conclusione, un Sys Admin?

- Installare tutti I sistemi operativi
- Conoscere tutti I linguaggi di programmazione
- Almeno l'inglese tecnico
- Risolvere I problemi in modo rapido e funzionale (o anche mettere pezze)
- Risolvere I problemi in modo elegante e funzionale (incrociando le dita di averne il tempo)
- Una fetta di culo no, eh?
- Deve conoscere ogni vite del sistema che amministra
- Dovrebbe creare e mantenere la documentazione sul sistema
- Deve conoscere alcuni tediosi articoli della legge italiana...
- Insomma deve sapere tutto (o quasi)

Capitolo 2

Hardware e RAID





La scheda madre di un computer è una scheda formata da chip (circuiti integrati elettronici) Contiene l'unità centrale di elaborazione (CPU), le memorie e gli slot (prese) per le schede di espansione con le quali possiamo collegare le periferiche.

Le memorie presenti in un computer possono essere suddivise in: **memorie centrali** o principali (main memory) **memorie di massa**, **memorie esterne** (USB).
La capacità Le memorie possiedono una capacità che si esprime in Byte Scala di equivalenza: 1 Byte equivale a 8 bit; 1 chilo Byte (1 KByte) equivale a 1024 Byte; 1 mega Byte (1 MByte) equivale a 1024 KByte; 1 giga Byte (1 GByte) equivale a 1024 MByte; 1 tera Byte (1 TByte) equivale a 1024 GByte. 1 peta Byte (1 PByte) equivale a 1024 TByte).

Le Memorie Centrali

- Le memorie principali: Vengono anche chiamate memorie centrali Contengono un numero limitato di informazioni. Si dividono in: memoria RAM (Random Access Memory), cioè memoria ad accesso casuale, consente sia la scrittura che la lettura dei dati in essa contenuti memoria ROM (Read Only Memory), cioè memoria di sola lettura, consente soltanto la lettura dei dati in essa contenuti memoria Cache.
- Memoria RAM: La RAM (Random Access Memory) è la memoria che contiene i dati e i programmi in corso di esecuzione E di tipo volatile, significa che perde il suo contenuto quando il computer viene spento.
- Memoria ROM: La ROM (Read Only memory) è una memoria di sola lettura. Contiene un programma che permette di accendere il computer, chiamato BIOS (Basic Input Output System).
- Cache memory: La memoria cache (termina che deriva dalla lingua francese e che significa nascosto) svolge un compito di memorizzazione temporanea dei dati. Coadiuva la CPU nella comunicazione con la memoria RAM.

Le Memorie di Massa Hanno lo scopo di conservare i programmi e i dati in modo permanente. Le memorie di massa più diffuse sono collocate all'interno del case (memorie interne) Le memorie di massa che possono essere collegate esternamente al computer, vengono chiamate memorie esterne.
Si presentano in moltissimi formati, che vanno dai meno recenti floppy disk fino ai nuovissimi blu-ray disk. Quasi tutti sono composti da dischi estraibili, eccezion fatta per i dischi fissi o dischi rigidi (hard disk).
Dischi Fissi Sono collocati all'interno del case e non sono normalmente estraibili né visibili dall'esterno I primi modelli avevano una capacità di pochi. MByte, mentre i modelli più attuali fino ad alcuni TByte Normalmente ogni PC ne contiene uno solo ma è possibile aggiungerne anche qualche altro.

La CPU. Durante il suo funzionamento, non fa altro che eseguire le istruzioni di un programma Il programma è formato da istruzioni scritte attraverso

un linguaggio apposito, chiamato linguaggio macchina, composto da istruzioni scritte in forma binaria. La CPU è in grado di eseguire milioni di istruzioni al secondo e, in base al tipo di istruzione, incarica altri dispositivi di eseguire alcuni compiti.
Se viene impartito al computer il comando di stampare un documento, la CPU incarica la periferica di eseguire la stampa, quindi i dati vengono letti dall'unità a disco e inviati attraverso il bus alla periferica desiderata. L'utente effettua pertanto solo un click e al resto penserà la CPU.

Architettura di un Host (Server) Come sarà più chiaro in seguito, la parola chiave quando si tratta di strutturare un server è "ridondanza". In generale, "più componenti sono ridondanti in un server, meglio è" è un'affermazione parecchio solida.
A livello visivo, il server è spesso composto da un armadio (rack) nel quale possono venire installati vari moduli attraverso delle "rotaie". Questi moduli sono, ad esempio, gruppi di continuità, unità esterne di computing, e ovviamente, l'unità centrale del server.
Un server, proprio per il fattore di ridondanza, può avere più di un alimentatore, più di un processore, e così via.
Oltre a queste caratteristiche di base, è utile confrontare l'hardware di un PC con quello di un server, in quanto le componenti necessarie per far funzionare entrambi sono le stesse, ma ci sono differenze sostanziali che influiscono sul funzionamento e sulle prestazioni dei sistemi:

- **CPU:** i processori per server (i.e.: Intel Xeon, ecc.) sono dotati di più cores e più memoria cache, ma tendenzialmente operano a frequenze più basse rispetto ai processori consumer-grade in quanto i processi server sono ottimizzati per l'esecuzione multicore, mentre i processi più comuni per gli utenti lavorano per la maggior parte su un singolo core, e di conseguenza fanno affidamento sulla frequenza di clock del processore.

- **RAM:** anche qui la differenza è marcata, in quanto le RAM consumer-grade (specialmente le RAM DDR4) fanno affidamento sulle frequenze alte ma sacrificando i tempi di latenza, mentre le RAM da server adottano un approccio che favorisce la latenza a scapito della frequenza. Inoltre, le RAM da server sono dotate di tecnologia ECC (error correction code), una tecnologia che permette di correggere gli errori presenti nei dati memorizzati.

- **Dischi:** i dischi sono molto più veloci, spesso anche il doppio rispetto a quelli consumer-grade. Inoltre vengono studiati per durare più a lungo, sono dotati di una memoria cache di relativamente grandi dimensioni e vengono disposti in RAID nella quasi totalità dei casi.

- **Schede di Rete:** di solito il PC che abbiamo a casa fa affidamento sulla scheda di rete integrata sulla scheda madre, mentre nel caso dei server solitamente si usano almeno due schede di rete dedicate con porte gigabit o fibra.

- **Raffreddamento:** mentre nei PC consumer grade il raffreddamento è affidato a un dissipatore (ad aria o a liquido), nei server anche la sala dove

sono posizionati deve essere refrigerata a causa della quantità di calore che viene sprigionata. Questo rende la sala server parecchio rumorosa.

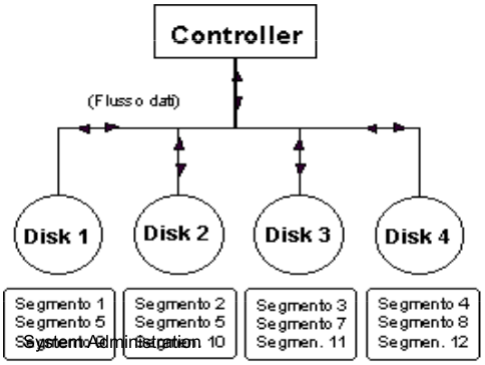
- **Alimentatori:** gli alimentatori per i server non solo hanno una capacità (in termini di Watt) superiore a quella che verrà effettivamente richiesta dal sistema, ma spesso un server ha più alimentatori nel caso in cui uno o più di questi falliscano (ridondanza).

- **Gestione remota:** esiste la possibilità di controllare remotamente il server grazie ad una tecnologia sviluppata da Hewlett-Packard, chiamata ILO (Integrated Lights Out). Fisicamente si presenta come una porta Ethernet dedicata. Le possibilità che si hanno grazie a ILO sono quelle di riavviare il server, accendere il server, montare disk images da remoto e accedere al server IML (Integrated Management Log), tutto questo grazie al fatto che la ILO è dotato di una connessione separata da quella del server.

Prima di parlare dell'organizzazione dei dischi, una piccola considerazione: per quanto possano aumentare le capacità di archiviazione dei dischi grazie all'addensarsi delle aree di memoria nei piatti, le velocità diminuiscono di pari passo. Una soluzione sarebbe quella di adottare degli SSD, ovvero delle unità a stato solido che memorizzano i dati grazie a delle porte NAND. Permettono di raggiungere velocità elevate in lettura e scrittura, ma sono meno affidabili e tendenzialmente hanno una durata di vita minore rispetto ai tradizionali hard disk.

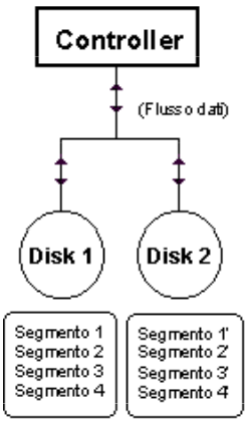
RAID - Redundant Arrat of Inexpensive/Independent Disks Il RAID non è altro che un costrutto logico (astrazione) che permette di organizzare due o più dischi in maniera tale da raggiungere migliori specifiche in termini di sicurezza, velocità e/o capacità. Questo costrutto può essere messo in atto grazie a un controller RAID, integrato nella maggior parte delle schede madri, ma per risultati migliori (specialmente in presenza di grandi quantità di dischi) si fa affidamento su controller RAID esterni collegati alla scheda madre tramite connettori PCI/E.
Esiste anche la possibilità di eseguire un RAID a livello software, ma in genere non viene utilizzato in quanto non è conveniente in termini di affidabilità e prestazioni. Questo è dovuto al fatto che il RAID software è un'astrazione ulteriore rispetto al RAID hardware, processo che avviene ad un livello più basso.
Prima di affrontare il RAID nelle sue forme principali, va precisato che questo **non** sostituisce il backup.

- **RAID 0:** detto anche striping, è un sistema in grado di separare i dati in singole stringhe in modo da scrivere ognuna di queste in un disco diverso.

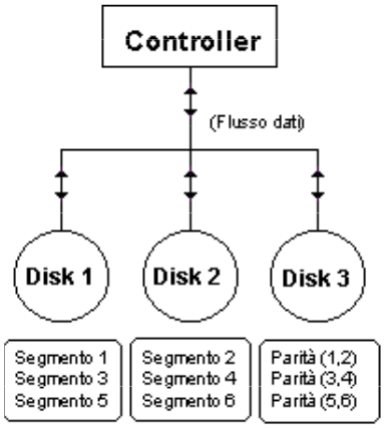


Migliora la velocità in lettura in base al numero dei dischi, ma è molto rischioso in quanto basta il malfunzionamento di uno qualsiasi dei dischi e i dati vengono persi.

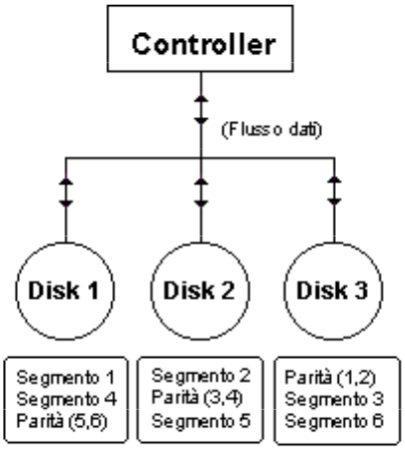
- **RAID 1:** detto anche mirroring, questo tipo di RAID mantiene una duplice copia degli stessi dati in due dischi diversi. Questo comporta all'avere una maggiore velocità in lettura, ma prestazioni pessime in scrittura, in quanto i bit dei dati vengono scritti più volte. Il vantaggio è quello di avere la massima ridondanza dei dati. Pur tenendo conto di quest'ultima caratteristica, le prestazioni ridotte lo rendono applicabile solamente nei piccoli sistemi o dov'è fondamentale la salvaguardia dei dati.



- **RAID 3** sistema simile allo striping: necessita di almeno tre dischi, ma in uno di questi vengono memorizzate le stringhe di parità, ovvero una specie di mappatura dei bit che permette la ricostruzione dei dati in caso di guasti. Questo però non permette al sistema di essere espandibile, in quanto il disco di parità non avrebbe la capienza per contenere tutte le stringhe. per espanderlo bisognerebbe cambiare tutti i dischi. Si hanno alte prestazioni in accesso a grossi files, ma letture e scritture non simultanee.

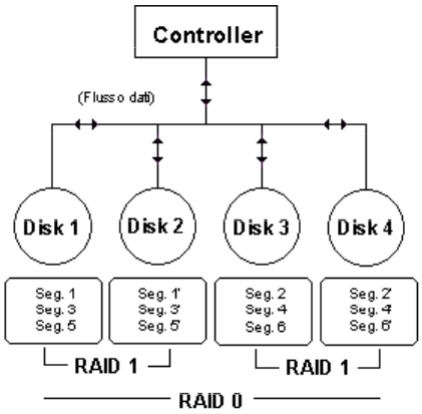


- **RAID 5:** è un'ottimizzazione del RAID 3, dove le stringhe di parità sono salvate in modo ordinato in tutti i dischi del RAID. Necessita di almeno tre dischi (tutti della stessa dimensione), e la capacità è sempre pari a `((n-1)*C)` byte, dove `n` è il numero di dischi e `C` è la loro capacità. Questo sistema è usato nei grossi sistemi multiutenza, dove le operazioni di scrittura sono molto inferiori a quelle di lettura. Il fault tolerance è pari a 1, ovvero il sistema funziona anche se un disco fallisce.



- **RAID 6:** questo sistema usa una divisione a livello di blocchi con i dati di parità distribuiti due volte tra tutti i dischi. Non era presente tra i livelli RAID originari. Nel RAID 6, il blocco di parità viene generato e distribuito tra due stripe di parità, su due dischi separati, usando differenti stripe di parità nelle due direzioni. Il RAID 6 è più ridondante del RAID 5, ma è molto inefficiente quando viene usato in un numero limitato di dischi.
- **RAID 7:** il RAID 7 è un tipo di RAID proprietario che aggiunge un sistema di caching ai RAID 3 o 4.
- **RAID 1+0:** sistema in grado di rigenerare i dati anche in presenza di rotture in più dischi, mantenendo buone prestazioni (migliori rispetto ai

raid 5 e 6 in quanto non devono venire gestite le informazioni di parità). Rimane un RAID molto costoso, in quanto la capacità complessiva dei dischi va sempre dimezzata, e l'espansione del sistema va effettuata con coppie di dischi uguali. Necessita di almeno quattro dischi per essere messo in atto. Viene impiegato in grossi sistemi dove si necessita di velocità massima e una sicurezza dati di alto livello.



Capitolo 3

Network Attached Storage - NAS

In una rete aziendale la flessibilità è molto importante: poter espandere la capacità di storage al crescere della rete senza però intervenire sui dispositivi già installati può essere un problema.
Fino a 10 anni fa era necessario acquistare nuovi server da aggiungere alla rete o in sostituzione di altri solo per avere più storage: DAS (Direct Attacched Storage) o SAS (Server Attacched Storage).
Ora tramite la tecnologia NAS si possono aggiungere dispositivi con grosse capacità di storage (e spesso scarse di calcolo) che garantiscono scalabilità nel tempo a costi contenuti. Il NAS è un computer (host) con:

- Almeno una CPU
- Una scheda madre
- `N` hard disk in raid (1,1+0,5,6 ecc)
- `K` SSD da utilizzare come cache (Sinology)
- Un sistema operativo (Linux based, solitamente)
- `M` schede di rete veloci (almeno 1Gbs)

Quali sono gli utilizzi del NAS?

- Si utilizza come storage per le home utenti .
- Si utilizza come storage per i backup.
- Si utilizza come storage per le macchine virtuali.
- Si utilizza come storage per database, siti web ecc.

Insomma è uno storage, ma va pesato a seconda degli usi e possibilmente rindondato.
Un NAS può contenere più volumi, che possono essere replicati su altri NAS e visibili solo da chi si vuole (tramite ACL).
La formula RAID + Rindondanza NON sostituisce il backup.

Capitolo 4

Storage Area Network - SAN

Il SAN è una soluzione altamente professionale adottata da quasi tutte le grosse compagnie (meno popolari nelle medie dimensioni) per fronteggiare le necessità di storage. Il ruolo del SAN è quello di sostituire quelli che sono i sistemi più tradizionali, come ad esempio DAS (Directly Attached Storage) o SAS (Server Attached Storage). In sostanza, consiste in una LAN dedicata unicamente allo storage, affiancata alle LAN aziendali per non impattare sulle loro prestazioni. Il vantaggio rispetto ai sistemi tradizionali è particolarmente evidente se si opera un confronto tra questi e il SAN.
Tradizionalmente, vengono utilizzati dispositivi collegati direttamente ai server che condividono le informazioni sulla LAN. Questo sistema presenta dei limiti:

- Gestione dell'accesso alle informazioni complicata.
- La rete è un collo di bottiglia (1000MBit).
- I server sono un collo di bottiglia.
- Il formato dei dati è vincolato al sistema a cui è connesso il device.
- Le prestazioni dei server e dei devices non sono pienamente sfruttate.
- Degrado delle prestazioni della rete.
- Inefficienza crescente al crescere delle richieste e degli utenti.
- Spreco di banda, CPU e RAM: ogni trasferimento implica la negoziazione di parametri per la connessione.

Il SAN, invece:

- Architettura scalare che si adatta bene alle evoluzioni di una azienda.
- Permette di collocare le risorse dove servono.
- Sono il risultato di un buon lavoro di progettazione.
- Permettono la scalabilità in termini di storage, banda e connettività senza interruzione dei servizi.

Il fattore della dislocazione geografica è da tener conto: infatti, se un'azienda ha più edifici il SAN si può applicare con un collegamento in fibra ottica se gli edifici sono in un raggio di 10km, con protocolli di Storage Over IP altrimenti.

Capitolo 5

Canali di Trasmissione

Mezzo o canale trasmissivo è il supporto fisico tramite il quale un segnale si propaga da un punto ad un altro di una rete.
Quando si mettono in collegamento due interlocutori si dice che fra questi si stabilisce un canale di comunicazione.
Su un mezzo trasmissivo vi possono essere simultaneamente più canali e un canale può usare più mezzi trasmissivi.
I mezzi e canali di trasmissione sono così classificati:

- **Mezzi di trasmissione:**
  - doppino telefonico
  - cavo coassiale
  - fibra ottica
  - etere (wi-fi)
- **Canali**
  - simplex: flusso dati unidirezionale
  - half-duplex: flusso alternativo nelle 2 direzioni
  - full-duplex: flusso simultaneo nelle 2 direzioni
  - punto-punto e multi-punto

I parametri rilevanti quando si parla di mezzi o canali di trasmissione sono i seguenti:

- **Velocità di trasmissione**
  - i dati sono codificati in bit
  - si misura in bps (bit per secondo) o con i suoi multipli Kbps (kilobit per secondo) e Mbps (megabit per secondo) e così via
- **Larghezza di banda**
  - indica la massima capacità trasmissiva di un mezzo
  - si misura anch'essa con le unità di misura indicate in precedenza

Confronto banda/velocità: su un mezzo trasmissivo possono essere realizzati più canali che "dividono" la banda; pur avendo allora un'elevata larghezza di banda, ciascun canale avrà una ridotta velocità di trasmissione.

Doppino telefonico (rame): è costituito da due a otto sottili fili di rame intrecciati, e il suo utilizzo varia da rete a rete, ognuna delle quali hanno categorie diverse di cavo, che definiscono la banda massima garantita. Inoltre, più si sale di categoria, maggiori sono le proprietà che acquisisce il cavo, come il miglioramento della qualità costruttiva, la minor perdita di segnale, ecc..

Fibra ottica: è costituita da decine o centinaia di sottili fibre di vetro o di materiale plastico che trasmettono impulsi di luce.
Già utilizzata per le dorsali oceaniche e per le LAN ad alta velocità. Un sistema di trasmissione su fibra ottica consiste di una sorgente luminosa, del mezzo di trasmissione e di un ricevitore; la sorgente luminosa converte segnali elettrici in impulsi per poi riconvertirli in segnali elettrici alla fine del tragitto. Diciamo pure che un segnale luminoso rappresenta il bit a 1 e l'assenza di luce rappresenta il bit a 0.
La fibra ottica necessita di ripetitori, per via dell'attenuazione del segnale, ogni 50km rispetto i 5km del doppino, non è affetta da disturbi esterni e ha una capacità di trasmissione molto elevata.

WiFi: viene usato l'etere come mezzo trasmissivo e consentono di creare facilmente reti senza installare alcun tipo di cablaggio, operazione in molti casi costosa.
Si utilizza un apparecchio, detto access point, collegato fisicamente alla rete, che comunica con gli utenti attraverso segnali radio. I computer degli utenti devono essere dotati di una scheda per il collegamento wireless, detta scheda wi-fi.

Capitolo 6

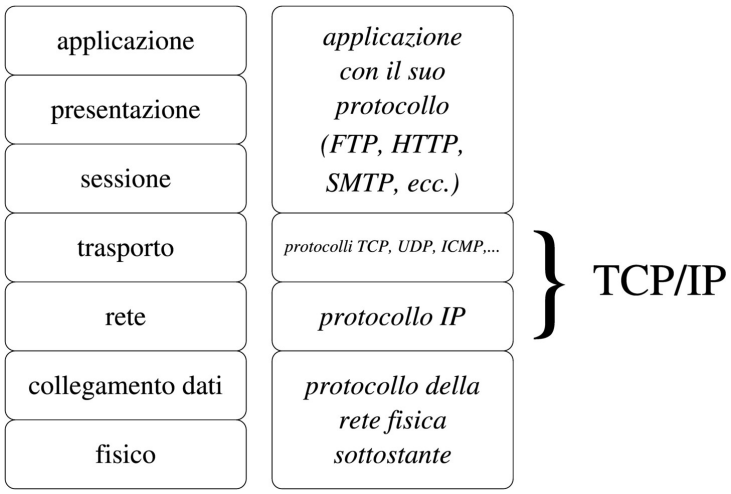
Networking

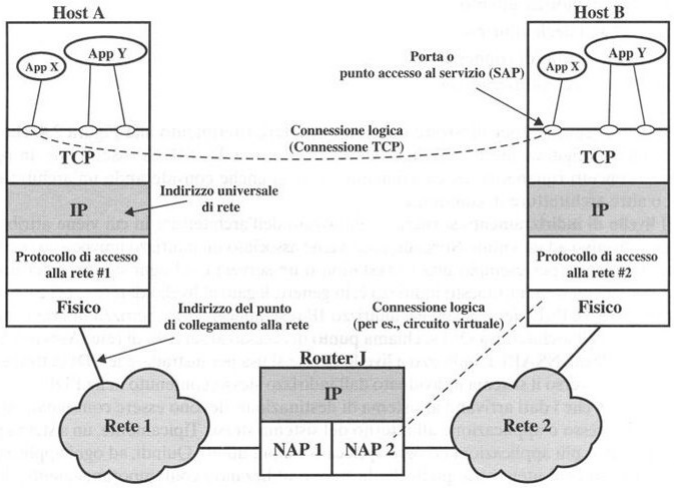
In alcune situazioni l'amministratore di sistema deve essere anche un po' amministratore di rete: ad esempio nelle piccole aziende i due ruoli coincidono.
Le reti possono essere classificate in tre categorie:

- **LAN o Local Area Network:** identifica una rete costituita da computer collegati tra loro, dalle interconnessioni e dalle periferiche condivise in un ambito fisico delimitato
- **MAN o Metropolitan Area Network:** è una rete dati che interconnette un'area corrispondente a quella di una grande città. Le reti di questo tipo vengono realizzate con tecniche innovative come, per esempio, la posa di cavi a fibre ottiche o la tecnologia wireless. Una MAN può interconnettere tra loro diverse LAN
- **WAN o Wide Area Network:** indica in generale una rete di grande estensione, e a cui in genere si fa riferimento per indicare la struttura che connette varie reti locali, estendendosi potenzialmente su tutto il mondo (Internet è un esempio di WAN)

Il protocollo TCP/IP: per protocollo si intende un insieme di regole che delineano il formato che deve essere utilizzato nella comunicazione tra due sistemi.
Il TCP/IP Internet Protocol Suite è il nome dato ad un set di protocolli sviluppato in seno al DoD degli Stati Uniti (Department of Defense) dal DARPA (Defense Advanced Research Project Agency).
All'epoca esisteva una sola rete, ARPA-net, che connetteva solo pochi computer a livello universitario. I dati, per transitare su di una rete, devono essere divisi in piccoli pezzi, ognuno dei quali viene inviato separatamente. Su di una rete IP, questi pezzetti sono chiamati "pacchetti"; tutti i trasferimenti di dati avvengono sotto forma di pacchetti.Il protocollo TCP/IP è costruito con un modello a strati, in cui le informazioni si fanno strada da un'applicazione su un host ad un'applicazione su un altro host attraverso varie trasformazioni (incapsulamenti, frammentazione, ecc).
In una rete TCP/IP ogni host è identificato da un indirizzo IP univoco all'interno di essa. Non rispecchia lo schema standard ISO/OSI per la scrittura corretta

di un protocollo di comunicazione ma è scritto talmente bene ed è talmente semplice da essere sempre efficiente.





I dispositivi che compongono la struttura di un network sono i seguenti:

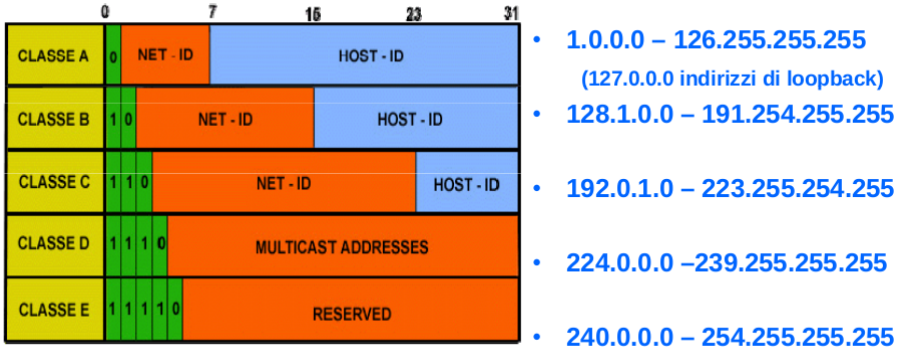
- **Router:** si occupa di instradare I pacchetti in/out rispetto alla LAN. Spesso è uno switch evoluto.
- **Hub:** inoltra I pacchetti su tutte le porte la velocità viene condivisa e si presentano collisioni.
- **Switch:** mantiene una tabella di corrispondenza tra porte e mac address delle schede di rete ad esse collegate. In tal modo instrada I pacchetti direttamente alla scheda di rete corretta, evitando collisioni e garantendo sempre la velocità massima.
- **Access point:** permette accesso alla rete tramite connessione wifi (è un po' router e un po' switch).

È interessante notare come in un semplice network casalingo alcuni di questi componenti sono presenti in un unico dispositivo, mentre in una rete di un certo spessore ogni dispositivo ha il suo compito.

IPv4: ogni qualvolta un dispositivo si connette a un network, gli viene assegnato un indirizzo IP. Gli indirizzi IP versione 4, , sono composti da una sequenza di 32 bit, suddivisi convenzionalmente in quattro gruppi di 8 bit, e rappresentati in modo decimale separati da un punto.
All'interno di un indirizzo del genere si distinguono due parti: l'indirizzo di rete e l'indirizzo indirizzo del nodo particolare. Il meccanismo utilizzato per distinguere la parte dell'indirizzo che identifica la rete è quello della maschera di rete o netmask.
La maschera di rete è un indirizzo che viene abbinato all'indirizzo IP da analizzare con l'operatore booleano AND, per filtrare la parte di bit che interessano.

Classi di indirizzi IPv4:

- **Indirizzi di classe A:** Il valore del primo ottetto è compreso tra 1 e 126. E' rappresentata da indirizzi di tipo: `Rete.Host.Host.Host` ovvero 8 bit per la identificare la rete (di cui il primo fisso) e 24 per identificare gli host. Permette di ottenere 126 reti formate da 16.774.214 host ciascuna.
- **Indirizzi di classe B:** Il valore del primo ottetto è compreso tra 128 e 191. E' rappresentata da indirizzi di tipo: `Rete.Rete.Host.Host` ovvero 16 bit per la identificare la rete(di cui i primi due fissi) e 16 per identificare gli host. E' possibile ottenere 16.384 reti formate da 65.534 host ciascuna.
- **Indirizzi di classe C:** Il valore del primo ottetto è compreso tra 192 e 223. E' rappresentata da indirizzi di tipo: `Rete.Rete.Rete.Host` ovvero 24 bit per la identificare la rete (di cui i primi tre fissi) e 8 per identificare gli host. E' possibile ottenere 2.097.152 reti con 254 host ciascuna.
- **Indirizzi di classe D:** Il valore del primo ottetto è compreso tra 224 e 239. Sono indirizzi di rete riservati ai gruppi multicast e non assegnabili ai singoli host.
- **Indirizzi di classe E:** Il valore del primo ottetto è compreso tra 240 e 255. Sono indirizzi riservati per usi futuri.



Esistono, infine, degli indirizzi "speciali":

- `0.0.0.0`: default route
- `127.0.0.1`: loopback
- Range classe A: `10.0.0.0` - `10.255.255.255`
- Range classe B: `172.16.0.0` - `172.31.255.255`
- Range classe C: `192.168.0.0` - `192.168.255.255`

Esistono due notazioni principali attraverso le quali è possibile indicare un indirizzo IP:

- **Indicando espressamente la subnet mask:**
  - `49.22.5.3 255.0.0.0` - Classe A;
  - `172.16.20.5 255.255.0.0` - Classe B;
  - `192.168.15.4 255.255.255.0` - Classe C;
- **Indicando i bit che compongono la subnet mask:**
  - `49.22.5.3/8` - Classe A;
  - `172.16.20.5/16` - Classe B;
  - `192.168.15.4/24` - Classe C;

Alcune classi di indirizzi, definite nella RFC 1918, vengono chiamati privati e sono utilizzati per le reti locali non connesse ad internet:

- Da `10.0.0.0.0` a `10.255.255.255.255`
- Da `172.16.0.0` a `172.31.255.255`
- Da `192.168.0.0` a `192.168.255.255`

Approfondimento sugli indirizzi speciali. Esistono alcuni particolari indirizzi di rete che non possono essere assegnati per l'identificazione di un host, tra questi abbiamo: network e broadcast e loopback:

- **Network:** quando i bit dell'ottetto che rappresenta l'host hanno tutti valore 0, l'indirizzo è detto di rete o Network Address: `192.168.5.0`
- `0.0.0.0`: quando tutti i bit hanno valore zero, identificano "questo host";
- **Broadcast:** quando i bit del numero che rappresenta l'host hanno tutti valore 1, l'indirizzo è detto di broadcast o broadcast address, e rappresenta tutti gli host di quella rete. Inviare un pacchetto all'indirizzo `192.168.5.255` equivale a mandare un pacchetto a tutti gli host della rete `192.168.5.0`;
- **Broadcast di rete:** abbiamo questo tipo di indirizzo quando tutti i bit, sia della parte relativa all'host sia della parte relativa alla rete hanno valore 1. Inviare un pacchetto a `255.255.255.255` significa inoltrarlo verso tutti gli host della rete corrente;
- **Loopback:** è utilizzato per funzioni di test del protocollo TCP/IP, non genera traffico di rete e corrisponde all'indirizzo `127.0.0.1`;

La struttura della codifica degli indirizzi IPv4 permette di avere un certo numero limitato di indirizzi IP codificabili. Questi indirizzi vengono assegnati a blocchi agli ISP, e ora sono finiti. È nata quindi l'esigenza di avere un nuovo formato di indirizzi IP, che permetta di codificare più unità.
Nascono quindi gli indirizzi **IPv6**.
Il nuovo formato dell'IP, ora di 128 bit a fronte dei 32 della precedente versione (IPv4), porta in sé notevoli cambiamenti che dovranno essere affrontati dall'intera utenza web.
Prima di IPv6 venne sperimentato il protocollo IPv5, che venne però abbandonato.
IPv6 usa indirizzi 4 volte più lunghi di quelli IPv4. IPv4 ha un massimo teorico di circa 4 miliardi di indirizzi, mentre IPv6 ha un massimo teorico di circa 340 miliardi di miliardi di miliardi di miliardi. Siccome gli indirizzi IPv6, come già quelli IPv4, devono essere strutturati per il routing e per altri scopi, il numero di indirizzi realmente utilizzabili è minore, ma sempre estremamente grande.
Un tipico indirizzo IPv6 è composto da 8 gruppi separati dal carattere ":". Ogni gruppo è composto da un massimo di quattro lettere e numeri, come può essere `2001:db8:1f70:999:de8:7648:6e8:1`.

Decrizione	IPv4	IPv6
Indirizzo	32 bit (4 byte) di lunghezza. L'indirizzo è composto da una parte relativa alla rete e da una parte relativa all'host, che dipendono dalla classe dell'indirizzo.	128 bit (16 byte) di lunghezza. L'architettura di base è a 64 bit per il numero di rete e a 64 bit per il numero host. Spesso, la parte host di un indirizzo IPv6 (o parte di essa) deriverà da un indirizzo MAC o da un altro identificativo dell'interfaccia. Il numero di indirizzi IPv6 è 10^28
Maschera	Utilizzata per distinguere la parte relativa allarete da quella relativa all'host.	Non utilizzata
NAT	Le funzioni firewall di base integrate in TCP/IP	Al momento, NAT non supporta IPv6

IPv4 è sopravvissuto fin'ora grazie al NAT che permette di convertire indirizzi di rete privati in un unico indirizzo di rete pubblico. (definzione molto approssimativa).
Perché, però, non passiamo direttamente agli indirizzi IPv6?
Per la stessa regola enunciata all'inizio del documento: perché cambiare un sistema che funziona?
Perché ad esempio un firewall con `iptables` (che vedremo) configurato per IPv4 viene totalmente bypassato da IPv6.
Perché significa dover adeguare apparati, sistemi operativi, host e personal computer e cambiare regole e politiche di sicurezza interne.

Capitolo 7

Filesystems

I filesystems sono la parte del sistema operativo che si occupa della gestione dei files, formattando opportunamente le unità di memoria di massa, registrando e leggendo i files.
Viene visto in modi differenti:

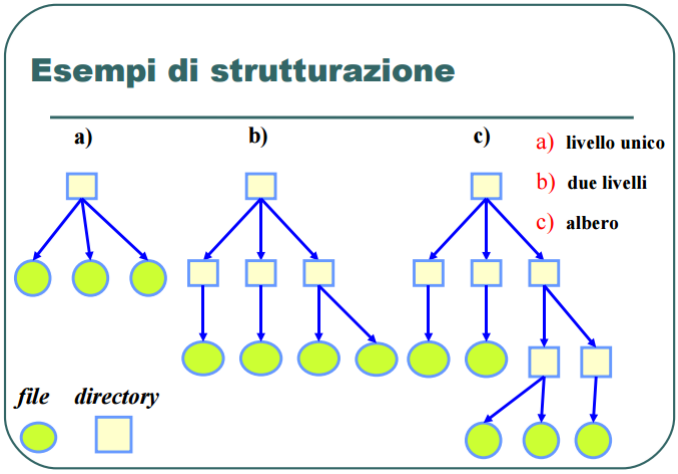
- **Utente:** appare come un insieme di file e directory ed è usato per memorizzare e organizzare i dati in modo che siano accessibili al sistema e ai suoi utenti.
- **OS:** è un insieme di strutture di controllo e blocchi di dati che occupano uno spazio ben definito da una partizione (porzione del disco) che permette di memorizzare e gestire i dati.

Che cos'è realmente un file? Un file non è altro che un'astrazione del sistema operativo che permette di usare in modo semplice ed efficiente i dispositivi di memoria secondaria.
Ogni file possiede attributi (nome, tipo, dimensione, protezione, data, ora ecc). I file possono essere creati, letti, scritti, cancellati ecc.. I nomi dei file possono avere, nei SO moderni una lunghezza compresa tra 1 e 255 caratteri (percorso compreso).

Esistono vari tipi di files:

- File regolari (o file utente, sono `ASCII` o `BIN`)
- Directory (contengono file regolari e non in una struttura gerarchica)
- File speciali a blocchi (unità di I/O a blocchi).
- File speciali a caratteri (unità di I/O a caratteri).
- I file si distinguono per le estensioni: `exe, jpg, gif, mov, mkv, ecc.`
- In windows un file speciale è quello di `Swap`

Directory: è un elenco di nomi di file (o/e altre directory) a cui è associato un nome. Fondamentale per dare una struttura gerarchica al filesystem. In una struttura ad albero la radice è detta radice del filesystem (/ o c:\). Sulle directory si possono effettuare le seguenti operazioni: creazione, cancellazione, link, unlink, rename ecc..



Struttura di un Filesystem in Linux/Unix

- `/etc`, directory che contiene, tipicamente, i file di configurazione del sistema.
- `/bin`, che contiene alcuni programmi essenziali.
- `/sbin`, che contiene alcuni programmi essenziali che tipicamente l'utente comune non deve lanciare autonomamente.
- `/lib` che contiene le librerie dinamiche necessarie ai programmi delle due directory precedenti.
- `/usr`, che contiene tutti i programmi che non sono necessari nelle prime fasi di avvio del sistema, nonché le relative librerie e file accessori, nelle directory `/usr/[s]bin`,`/usr/lib` e `/usr/share`, rispettivamente
- `/usr/local`, che dispone anche di sotto-directory simili a quelle di `/usr`, in cui vengono memorizzati programmi e librerie installati dall'amministratore e scelti al di fuori di quelli forniti dalla distribuzione. Funzione simile ha `/opt`, che viene usato per quei programmi che non rispettano la convenzione di separazione dei binari dalle librerie e dai file accessori.
- `/home`, contiene gli spazi di memorizzazione personali (home directory) degli utenti.
- `/root`, è la home directory dell'utente root.
- `/var`, contiene i dati di tutte quelle applicazioni di uso generale del sistema, e non del singolo utente. Qui dentro vi sono, tipicamente, i dati relativi ai server (software) che forniscono i vari servizi. Di particolare importanza

la directory `/var/log`, che contiene i log (registri degli eventi) del sistema, gestiti direttamente dalle relative applicazioni o dal servizio syslog.

- `/proc`,
- `/dev`, contiene file che rappresentano i dispositivi presenti nel sistema, come ad esempio dischi e terminali.
- `/tmp` e `/var/tmp` servono a memorizzare file temporanei.

Alcune directories più recenti aggiunte al Filesystem Linux sono:

- `/boot`, che contiene alcuni dei file necessari al gestore d'avvio (boot loader), nonché il kernel ed alcuni file di supporto.
- `/proc`, contiene dei file (o directory) che rappresentano i processi in esecuzione nel sistema: contiene una directory per ogni processo in esecuzione, e altri file che rappresentano lo stato del sistema. Ad esempio il file `/proc/cpuinfo` contiene informazioni sui processori presenti nel sistema, mentre la directory `/proc/sys`. contiene directory e file con cui è possibile modificare alcuni parametri del sistema operativo a tempo d'esecuzione (equivalentemente al comando sysctl).
- `/sys`, si occupa di rappresentare altre configurazioni a tempo d'esecuzione di alcuni aspetti del sistema collegati all'hardware. Possiede una struttura più regolare rispetto a /proc.
- `/run` si occupa di mantenere alcuni piccoli file di stato delle applicazioni di sistema che, sebbene siano necessarie anche per la cooperazione tra queste, non è necessario mantenere tra i riavvii. Tradizionalmente questo compito era assolto dalla directory `/var/run`.

Ecco un elenco dei Filesystems più conosciuti:

- DOS e DOSLike (Windows 95/98/Me): FAT, FAT16, FAT32.
- Windows NT,2000,...,2016: NTFS e le sue evoluzioni.
- Linux: minix, ext2, ext3, ext4, reiserFS, XFS, UFS, ZFS, JFS.
- MAC OS X: HFS, HFS+, APFS.
- Novell Netware.

Uno dei Filesystems più iconici è il FAT32:

- Ultimo sviluppo della vecchia FAT.
- Si basa sulla tabella FAT(File Allocation Table).
- Facilità di creare driver che lo ha portato ad essere l'unico filesystem che gira su tutti i sistemi (compresi i vecchi DOS).
- Ormai lo trovate solo sulle chiavette.
- Poco affidabile, lento e frammentario (caro vecchio defrag...).

- Per sicurezza la FAT è replicata in più zone del disco.

Visti i grandi svantaggi che presentava il FAT32, si è passati all'NTFS, filesystem sviluppato da Microsoft per tutti i sistemi da Windows NT in poi. Eccone alcune specifiche:

- Permette di gestire 4 miliardi di file e si basa su una tabella chiamata MFT (Master File Table).
- Rimane lento e obsoleto.
- Comunque sicuro, stabile e flessibile.
- Per sicurezza la MFT è replicata in più zone del disco.

Il filesystem di default per i sistemi Linux più recenti è Ext4, una versione migliorata di ext2. È un filesystem Journaled, veloce e performante anche per grandi quantità di piccoli files, ed è retrocompatibile con le versioni precedenti del filesystem.
Per i sistemi Apple, invece, il filesystem principale è HFS+, una versione Journaled di HFS. Rimane un filesystem vecchio e obsoleto, poco performante ma integrato alla perfezione con il sistema operativo, che ne copre i difetti. È stabile e affidabile, ma quest'ultima caratteristica viene meno con l'aumentare del "peso" del sistema operativo.
Tutto sto "journaled"... Ma cos'è? Il **journaling** è una tecnica utilizzata dai filesystems per preservare l'integrità dei dati da eventuali cadute di tensione. Ogni modifica ai dati, infatti, viene scritta in un file di log e solo secondariamente scritta sul disco.

Poiché i sistemi ispirati ad UNIX hanno una singola gerarchia di memorizzazione, non esiste il concetto di lettera di unità come nei sistemi Windows. L'estensione della capacità del file system avviene perciò tramite un'operazione detta montaggio (comando mount, file di configurazione `/etc/fstab`), in cui il file system di un'unità viene innestato in una directory del sistema. Ad esempio la directory `/home`, con tutte le sue sotto-directory, potrebbe risiedere su un disco (o una partizione) a sé stante. Per verificare quando spazio disco è occupato sui vari file system montati, si può usare il comando `df`.
Quando inserite una chiavetta USB in un sistema Linux, ad esempio Ubuntu, il sistema monta automaticamente il volume presente sulla chiavetta in una directory temporanea:

```
$ mount /dev/sdd1 on /media/pippo/FedoraWSLive2513 type iso9660
(ro,nosuid,nodev,relatime,uid=1000,gid=1000,iocharset=utf8,mode
=0400,dmode=05 00,uhelper=udisk
```

Come si crea, quindi, un filesystem in Linux/Unix?
Per creare un file system (nella seconda accezione descritta sopra, ovvero un'organizzazione fisica dei dati su disco) si ricorre ai comandi della famiglia `mkfs.*` sul device file appropriato. Si noti che solitamente, quest'ultimo, non indica l'intero disco, ma piuttosto una partizione o, con una gestione più sofisticata, un volume logico.

```
# mkfs.ext4 /dev/sdd1
# mkfs.ntfs /dev/sda1
```

Per creare partizioni su disco si possono usare, a seconda del tipo di partizionamento, `[c]fdisk` o `[g]parted`.
La gestione dei volumi logici e dei sottostanti gruppi di volumi e volumi fisici tramite lvm è più complessa, e la loro trattazione richiederebbe troppo tempo, ma è generalmente consigliata, sopratutto in ambito server dove può capitare di dover ridimensionare al volo porzioni del disco.

Capitolo 8

Boot di un Sistema Operativo

BIOS: è un programma scritto in una memoria permanente (una volta erano ROM poi EPROM, poi EEPROM) e controlla la prima fase del processo di avvio:

- Effettua un test del sistema (ad esempio un blando memtest), cerca e controlla le varie periferiche, ed infine cerca una unità di massa da cui avviare il sistema (hd, usb pen, dvd ecc).
- In particolare cerca un MBR (master boot record) di solito memorizzato nel primo settore, ne carica il contenuto in memoria e gli passa il controllo delle operazioni.

UEFI: supera le limitazioni del vecchio BIOS consentendo, tra l'altro il boot da dischi particolarmente capienti (di capacità superiore ai 2 Terabytes) utilizzando GPT (GUID Partition Table). GPT è parte integrante dello standard UEFI e porta con sé tutta una serie di novità rispetto all'utilizzo del MBR (Master Boot Record).
Ecco alcuni vantaggi di GPT:

- architettura indipendente da CPU e driver.
- ambiente preOS (accessibile cioè all'infuori del sistema operativo, prima della fase di boot di quest'ultimo) molto più completo e versatile rispetto al passato. Si pensi che di solito UEFI offre anche il supporto della connettività di rete.
- design modulare.
- eliminazione della necessità di un bootloader (fatta eccezione per utilizzi più avanzati).
- esecuzione di moduli firmati (funzionalità Secure Boot).

MBR, o Master Boot Record, è una tabella di partizioni che cerca la prima partizione (porzione del disco) attiva (con flag di boot = 1). MBR legge il record

di avvio che contiene le istruzioni su come caricare il boot loader per avviare il sistema operativo; successivamente carica il boot loader che assume il controllo del processo di avvio.

Il Boot di Windows: il processo di boot loader viene svolto dal file NTLDR che controlla il processo di selezione del sistema operativo e il rilevamento dell'hardware prima dell'inizializzazione del kernel. NTLDR (il boot loader di Windows) può visualizzare un menu da cui si può selezionare il sistema operativo operativo. La videata è basata sulle informazioni che si trovano nel file `boot.ini`, un semplice file di testo nascosto e in sola lettura.

Il Boot di Linux: Linux ha a disposizione tre bootloader (principali):

- LiLO (LinuxLoader)
- GRUB
- LoadLin

LiLO: è il kernel loader di Linux ed è composto da due parti: una avviabile e una che provvede ad installarlo. La parte che provvede ad installare LilO si appoggia sul file `lilo.conf`.

GRUB: il GRand Unified Bootloader è il bootloader più gettonato da chiunque installi un sistema operativo Linux based. Graficamente si presenta come un menù da quale scegliere che sistema operativo avviare.
In generale, GRUB si configura tramite un file presente in `/boot/grub` di nome `menu.lst`. Per installare GRUB, basta usare il comando

```
#grub-install /dev/sda
```

Su Ubuntu è un po' diverso: trovate le opzioni di default nel file `/etc/default/grub`. Altre configurazioni le trovate in `/etc/grub.d/`. Qui trovate diversi files, simili a script, che potete disattivare semplicemente togliendo i permessi di esecuzione. Per fissare le modifiche, usate il comando

```
#update-grub
```

Caricamento del kernel: dopo aver scelto un'opzione del boot loader il sistema carica in memoria il kernel del sistema operativo. Il kernel si occupa di controllare che periferiche ci sono nel computer, carica i driver relativi, effettua delle operazioni di check preliminari, riconosce le partizioni e cede il controllo al processo di init. è il primo processo che il kernel manda in esecuzione dopo che il computer ha terminato la fase di bootstrap. Esso ha il compito di portare il sistema in uno stato operativo, avviando i programmi e servizi necessari. Dato che init è sempre il primo processo eseguito, esso ha tipicamente il PID 1.

Init e Runlevel. Runlevel è un particolare configurazione software del sistema che permette l'esistenza di un solo gruppo di processi: è uno stato del sistema che permette solo determinate azioni. Vi sono sette runlevel, numerati da 0 a 6. Generalmente quando siete in idle su ubuntu, siete in runlevel 5, 3 se non è stata caricata l'interfaccia grafica.
Da root potete cambiare runlevel con il comando init e controllare in che livello siete con il comando runlevel:

```
#runlevel
#init 3 - passa a runlevel 3
```

Processo di init: `SysV init` era il processo standard nel mondo Linux per controllare l'esecuzione del software all'avvio:

- Il kernel cerca init nella directory `/sbin` e lo avvia cedendogli il controllo della fase di avvio.
- init avvia lo script `/etc/rc.d/rc.sysinit`.
- `rc.sysinit` gestisce quasi tutti i processi del loader di avvio ed esegue rc.serial.
- init esegue tutti gli script per il runlevel di default.
- init esegue `/etc/rc.d/rc.local`.

Il processo di init in dettaglio:

- Il kernel individua init nella directory `/sbin` e lo esegue cedendogli il controllo della fase di avvio.
- Appena eseguito init diventa padre di tutti i processi avviati automaticamente dal sistema. Tra i processi avviati vi sono l'attivazione dello swap, il check dei dischi ecc ecc.
- Init poi esegue lo script inittab che descrive la configurazione di ogni runlevel.
- Infine init avvia tutti i processi in background necessari al sistema cercando nella directory `/etc/rcX.d` dove `X` è il runlevel predefinito di inittab. In particolare termina tutti gli script il cui nome comincia con `K` e avvia quelli il cui nome comincia con `S`. Dopodichè si preoccupa di avviare i processi delle console di sistema (getty).

Ecco un esempio di inittab:

```
# inittab This file describes how the INIT process should
# set up the system in a certain runlevel.
#[...]
# Default runlevel.  The runlevels used by RHS are:
# 0 halt (Do NOT set initdefault to this)
# 1 Single user mode
# 2 Multiuser, without NFS (The same as 3, if you do not have
networking)
```

```
# 3 Full multiuser mode
# 4 unused
# 5 X11
# 6 reboot (Do NOT set initdefault to this)
```

Partizionamento: una partizione è una porzione del disco. Un disco può contenere diverse partizioni a seconda delle limitazioni del sistema operativo che lo utilizza. Ad esempio nei sistemi PC-Intel (MBR) il limite è costituito da 4 partizioni primarie. Una partizione primaria speciale è quella definita estesa, che può contenere fino a 16 unità logiche. In Linux questo limite è superato usando LVM.
Ora con lo schema di partizionamento GPT questi limiti dovrebbero essere superati.
A seconda del tipo di installazione che si deve effettuare è bene partizionare il disco in modo diverso.

L'area di swap: l'area di swap è una partizione che deve essere grande 1,5 volte (almeno) la dimensione della RAM installata nel sistema. Perché?
Storicamente conteneva l'intero dump della memoria in caso di crash del kernel, quindi al riavvio dopo il crash non c'era più spazio swap disponibile.
Windows gestisce lo swap tramite un file (pagefile.sys), che dovrebbe, in una installazione server, essere posizionato in una partizione diversa da C:.
Di seguito, gli esempi di partizionamento di due tipi di server (è stata omessa l'area di swap, ma va sempre creata!):

- Server web:
  - `/` massimo 50GB
  - `/var` almeno 50GB se c'è un piccolo DBMS
  - `/var/www` almeno 50GB se ci sono 10/20 siti
  - `/var/log` almeno 10GB
- Server mail:
  - `/` massimo 50GB
  - `/mailstore` almeno 100GB per 20 utenti
  - `/var` almeno 50GB
  - `/var/log` almeno 10GB
- Server home:
  - `/` massimo 50GB
  - `/home` almeno 100GB per 20 utenti
  - `/var` almeno 50GB
  - `/var/log` almeno 10GB
- Server database:
  - `/` massimo 50GB

- `/var` anche più di 100GB, può ospitare uno o più DBMS
- `/var/log` almeno 10GB

`/var/log` contiene i log di sistema; in caso di attacchi, malfunzionamenti o altro si può riempire.

Capitolo 9

La shell

La shell è un programma che gestisce la comunicazione tra l'utente e il sistema operativo (vero soprattutto in ambienti Linux/Unix prima dell'avvento delle interfacce grafiche). È un interprete dei comandi: un vero e proprio ambiente di programmazione con cui realizzare script per automatizzare operazioni complesse ripetitive.

Tipi di shell in Unix/Linux e OS X:

- Bourne Shell (sh): shell originale di Unix. Scritta da Steve Bourn, presente su tutti i sistemi unix.
- C Shell (csh): scritta all'università di Berkley California. Usa un linguaggio di scripting simile al C.
- TC Shell (tcsh): evoluzione di csh. Shell preferita di Sartoretto.
- Korn shell (ksh): scritta da David Korn è un miscuglio fra csh, tcsh e sh.
- Bourne Again Shell (bash): Scritta da FSF come shell free software. Shell di default di linux, usa un linguaggio compatibile con sh.
- Dash: deriva da Ash shell del sistema NetBSD.
- Rbash: restricted bash shell.
- Informazioni sulle shell installate nel sistema in `/etc/shells`.

Tipi di shell in Windows:

- Prompt dei comandi (cmd.exe): storico porting del vecchio DOS. Permette l'esecuzione di comandi di sistema e la creazione di script in batch (`.bat`). Da qualche anno permette l'esecuzione di Vbscript.
- PowerShell: noto inizialmente come Microsoft Shell o MSH (o col nome in codice Monad) e poi come Windows Shell è una shell caratterizzata dall'interfaccia a riga di comando (CLI) e da un linguaggio di scripting, sviluppata da Microsoft. È basata sulla programmazione a oggetti e sul framework Microsoft .NET. Permette l'esecuzione di script Vbs.

Un esempio concreto del funzionamento della bash si ottiene se pensiamo a cosa succede durante il login in modalità testuale:

- Cerca il file `/etc/profile` e, se esiste, lo elabora.
- Cerca il file `$HOME/.bash_profile`
- Se non esiste, allora cerca `$HOME/.bash_login`
- Se non esiste neanche questo, allora prova con `$HOME/.profile`.

Il file `.profile` è un file di configurazione generale contiene la definizione delle variabili d'ambiente.
`.bashrc` contiene comandi, alias e dichiarazioni di ulteriori variabili.
I file `/etc/profile` e `~/.bash_profile` contengono le configurazioni per una shell di login.
`$HOME/.bashrc` per quelle non di login.
In pratica i primi file vengono letti ed eseguiti solo per la prima shell che si apre, e non per le successive le quali eseguono ogni volta i `.bashrc`. La shell è in grado di ricordare i comandi immessi dall'utente e normalmente sono salvati nel file `~/.bash_history`, con proprietà `HISTSIZE, HISTFILE, HISTFILESIZE.`
Esempio:

```
# export HISTFILE=/dev/null
```

Caratteri speciali della bash:

- ; separatore comandi.
- & esecuzione in background.
- `()` raggruppa I comandi.
- {} blocco di comandi.
- `|` pipe.
- `>< &` reindirizzamento.
- `*?[] !` Caratteri jolly.

Reindirizzamento della bash:

- `0` stdin tastiera
- `1` stdout terminale
- `2` stderr terminale
- `<` file legge input da file
- `>` file scrive output su file creando/sovrascrivendo
- `>>` file scrive output su file creando/accodando se esiste
- `<<` text legge stdin fino a trovare una riga uguale a text

- `n<` file Associa al file il file descriptor n
- `n>` file Dirige l'output proveniente dal file descriptor n nel file
- `>&n` Duplica lo standard output nel file descriptor n
- `<&n` Duplica lo standard input dal file descriptor n
- `&>file` Dirige lo standard output e lo standard error nelfile
- `<&-` Chiude lo standard input
- `>&-` Chiude lo standard output
- `n>&-` Chiude l'output proveniente dal file descriptor n
- `n<&-` Chiude l'input proveniente dal file descriptor n

Comandi bash:

- Oltre ai comandi di sistema (`ls, df, cd, rm` ecc.) potete usare dei comandi propri della bash nei vostri script.
- `#!shell` indica la shell usata nel vostro script deve essere la prima riga (esempio: `#!/bin/bash`).
- `. file` o `source file` legge le righe di comando contenute in file.

Variabili in bash:

- Il carattere $ come prefisso per indicarne il valore.
- Si assegna il valore tramite il carattere `=`:
  ```
  MYVAR="PIPPO" # occhio agli spazi!!!
  ```
- Generalmente le variabili sono visibili solo all'interno della shell stessa. Per passare variabili ad altri programmi chiamati all'interno della shell si deve utilizzare il comando export.
- Se racchiuse da `[ ]` sono considerate come variabili di array.
- $var il valore della variabile
  ```
  # echo $MYVAR
  PIPPO
  ```
- `$0` nome del programma
- `${n}` singoli argomenti della riga di comando `n=1..9`
- `$#` numero argomenti riga di comando
- `$*` tutti gli argomenti della riga di comando
- `$$, $?, $!`

Capitolo 10

Utenti e gruppi

I sistemi simil-UNIX, tra cui Linux, sono generalmente multi-utente. Gli utenti posseggono un nome utente (login name), un identificativo numerico (user id o UID), un gruppo primario (vedremo poi a cosa serve), una home directory, una login shell ed alcune informazioni aggiuntive, come ad es. il nome reale. È importante notare che ciò che realmente identifica un utente è il suo UID.

Comandi utili per la gestione di utenti/gruppi:

- `whoami`: ottenere informazioni sul proprio username.
- `w`: ci dice che è collegato al sistema.
- `finger <user>`: da informazioni dettagliate su un utente.
- `id <user>`: restituisce l'uid dell'utente e dei gruppi di cui fa parte.

I gruppi, come indica il nome, raccolgono un insieme di utenti, e vengono usati per assegnare dei permessi garantiti collettivamente a tutti i membri del gruppo. Col comando `groups` è possibile vedere di quali gruppi si fa parte.

In UNIX esiste un utente speciale, avente UID 0 e, generalmente, il nome root, che è sostanzialmente onnipotente, ed è usato dall'amministratore di sistema. Poiché da grandi poteri derivano grandi responsabilità, l'amministratore usa un utente personale, e solo quando deve compiere operazioni che richiedano tali superpoteri, diventa root tramite il comando su o tramite sudo. Mentre il comando su apre una shell eseguita come utente root chiedendo la password di quest'ultimo, il comando sudo, eventualmente seguito da un altro comando da eseguire come root, chiede la password dell'utente di partenza, a patto che questo sia in una lista/gruppo speciale (admin, sudo, sudoers ecc).
Il modo in cui, in UNIX, si concede o si nega l'accesso a parti del filesystem a specifici utenti o gruppi è l'uso dei permessi. Ogni oggetto del filesystem (file, directory) ha, infatti, associati un utente ed un gruppo che posseggono il file, nonché almeno 9 bit di permessi, più una tripletta aggiuntiva di modificatori. Si distinguono i permessi di lettura (r),scrittura (`w`) ed esecuzione (`x`) per ognuna delle classi utente proprietario (u), gruppo proprietario (`g`) e altri (`o`). I modificatori, sono, invece, `setuid`, `setgid` e `sticky bit`.

Per cambiare il proprietario e il gruppo di un file si usano i comandi `chown` e `chgrp`, mentre per cambiare i permessi si usa il comando `chmod`.
In tempi più recenti sono stati proposti molti miglioramenti a questo sistema di permessi, sia tramite l'uso di attributi estesi sia tramite modelli di controllo alternativi della sicurezza (ad esempio quelli basati su ruoli), che permettono una granularità molto più fine nel controllo dell'accesso ai file (e non solo). Tuttavia, ad oggi, non esiste un completo consenso su un modello da adottare, e pertanto il minimo comune denominatore rimane sempre il sistema dei permessi.
NB: Windows utilizza un sistema di permessi molto più complicato basato su ACL che possono essere applicate a gruppi o utenti singoli. Oltre all'interfaccia utente, per assegnare permessi si possono usare i comandi `calcs` e `icacls`.

Quota. In sistemi condivisi è possibile assegnare ad ogni utente o gruppo di utenti una quantità massima di spazio disco occupato detta QUOTA.
Di seguito, alcuni comandi per la gestione della quota:

- `quota nomeutente` restituisce la stato
- `edquota u <utente>` permette di cambiare la configurazione della quota di un utente
- `edquota p <prototipo> <utente>` assegna lo schema di quota dell'utente prototipo a utente

In UNIX ogni programma in esecuzione è rappresentato da almeno un processo.
I processi osservano una rappresentazione virtuale della memoria, che ne permette l'isolamento.
Nel momento in cui viene lanciato, il processo assume (a meno dell'intervento dei modificatori `setuid` o `setgid` visti sopra) un utente ed un gruppo proprietario che sono quelli di chi ha eseguito il comando, e ne assumono i diritti di accesso.
Ogni processo è identificato da un PID (process id).
Per vedere la lista dei processi attivi si usa (di solito con vari parametri) il comando `ps`. È inoltre possibile vedere la genealogia dei processi (chi ha lanciato chi) tramite il comando `pstree`. Una visione più dinamica dei processi in esecuzione si ha tramite il comando top, o la sua alternativa htop.
È possibile mandare un segnale o uccidere un processo tramite il comando `kill` o `killall`.
È inoltre possibile modificare vari tipi di priorità di un processo usando i comandi:

- `nice`: lancia un programma con una determinata priorità.
- `renice`: cambia la priorità di un processo in esecuzione.
- `ionice`: cambia la classe di I/O scheduling (idle, realtime, best-effort) e la priorità del processo.

È possibile ottenere altre informazioni sullo stato del sistema, oltre che dai già visti comandi `df`,`[h]top` e `cat /proc/cpuinfo`, dai comandi `free` (uso della memoria), `iotop` (visione dinamica dell'uso dei sistemi di input/output), `vmstat`

(informazioni varie su memoria virtuale e i/o), `dmesg` (ultimi messaggi del kernel), `lspci`,`lsusb`,...
Un comando molto utile è `lsof`, che mostra i file aperti dai processi, rendendo possibile identificare quali programmi stanno, ad esempio, bloccando l'operazione di smontaggio di un filesystem.

Capitolo 11

Gestione della Rete

È possibile vedere le connessioni di rete (o tramite unix socket) effettuate dai processi in esecuzione, nonché altre informazioni, tramite il comando `netstat`.
La gestione delle interfacce avviene tramite il comando `ifconfig` e quella delle tabelle di instradamento tramite il comando `route`.
Nei sistemi Debian/Ubuntu il file di configurazione principale per questi aspetti è `/etc/network/interfaces`, che vengono usati dai comandi `ifup` e `ifdown`.
Per quanto riguarda la configurazione dei domain name server (DNS) da consultare, ci si riferisce al file `/etc/resolv.conf`, che però può anche essere generato dinamicamente a partire dalle configurazioni viste sopra.
`traceroute` stampa il percorso che fanno I pacchetti dall'host su cui è lanciato fino all'host di destinazione.
In Windows, la maggior parte delle operazioni si fa tramite interfaccia grafica. In particolare:

- Active Directory Users and Computers: per la gestione del catalogo utenti.
- DNS: per la gestione di un server dns.
- Share: per la condivisione windows di file e cartelle.
- Event viewer: per la consultazione dei log di sistema.
- File explorer: utile per la gestione dei permessi.

Ci sono però delle eccezioni, come i comandi `ipconfig, netsh, ping, tracert`.

Capitolo 12

Servizi di Rete

Tradizionalmente, i sistemi UNIX, dopo l'avvio del kernel, lanciano il comando init, che si occupa di gestire tutte le successive fasi di avvio, compresi il montaggio di file system aggiuntivi e l'avvio di programmi per fornire servizi. Si occupa infine di lanciare i gestori dei login da terminale.
Esistono molte varianti di questo programma, ma quella tradizionalmente più usata nei sistemi Linux prevede che il sistema passi attraverso una serie di stati (o runlevel), all'inizio e al termine dei quali vengono eseguiti degli script, residenti in `/etc/init.d`) che si occupano di lanciare o uccidere determinati processi.
Recentemente questo sistema è stato spesso rimpiazzato da diversi altri, tipicamente basati su eventi, a seconda della distribuzione (ad es. upstart). Tuttavia al momento il consenso sembra essere quello di adottare per il futuro il sistema systemd.
Tutti i vari sistemi descritti sopra vengono astratti, per quanto riguarda l'avvio e lo spegnimento manuale di servizi, dal comando service.

```
# service nfs-kernel-server restart
```

Tra i servizi lanciati dal sistema init (o simili) vi può essere un gestore delle operazioni periodiche, che negli Unix prende il nome di **cron**. Questo sistema permette di lanciare periodicamente comandi, specificando anche l'utente con cui debbano essere eseguiti.
Per operazioni programmate una tantum si può invece usare il comando `at` (`batch`, `atq`, `atrm`...).
Nei sistemi UNIX, il demone inetd è spesso il modo più semplice di mettere in opera un servizio. Questo "superserver", infatti, resta in ascolto sulle porte configurate, e, qualora riceva una richiesta (di connessione nel caso di protocolli stream-oriented, come TCP, o un semplice pacchetto nel caso di protocolli datagram-oriented, come UDP), si preoccupa di lanciare il programma desiderato, nonché di reindirizzare standard input, output ed error dello stesso sulla rete.
Vi sono diverse implementazioni di questo demone, da quelle più tradizionali (ad es. nel pacchetto Debian/Ubuntu openbsd-inetd) a quelle che aggiungono funzionalità, ma che presentano modalità di configurazione più complesse (ad es. nel pacchetto xinetd). Nel primo caso la configurazione risiede nel file `/etc/inetd.conf`.
Spesso abbinato all'uso di inetd è quello del tcp wrapper tcpd, che si occupa di

registrare le richieste in entrata, nonché di accettarle o rifiutarle in base ad una serie di semplici regole, definite nei file `/etc/hosts.allow` e `/etc/hosts.deny`. Tra i servizi di base possiamo elencare `ftp`, `tftp`, `telnet`, `rsh`.

NTP. All'interno di un sistema distribuito è fondamentale che client e server siano sincronizzati, per autenticazione, condivisione, sincronizzazione, esecuzione di job periodici ecc..
Il protocollo che ci permette di mantenere data e ora sincronizzati si chiama NTP (Network Time Protocol).
NTP permete ai client di sincronizzarsi con dei server, chiamati Time Server, che possono essere collegati ad un dispositivo di misurazione del tempo, quale un orologio atomico o un ponte radio, ad una stazione di misurazione, oppure ottenere le informazioni temporali da un altro server NTP.
I dispositivi di misurazione diretta del tempo (orologio atomico) vengono detti strato `0`, mentre i server NTP direttamente collegati ad essi vengono detti server allo strato `1`, o primari. I dispositivi che ottengono le informazioni da uno o più server allo strato n sono allo strato `n+1`.
Per la configurazione del NTP è sufficiente editare il file `/etc/ntp.conf` inserendo nella direttiva `server` il server ntp di riferimento.

Altri comandi utili sono `rdate`, che permette la sincronizzazione con un server che abbia a disposizione il servizio `time`, e `hwclock`, che scrive l'ora del sistema operativo sul bios del computer. Attenzione però, funzionano solo da root.
E Windows Server? Non dispone in modo automatico di questo servizio, per cui è necessario attivarlo manualmente modificando una chiave di registro tramite `regedit`: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentContr olSet\services\W32Time\Config)`.
Dopodiché, dovete andare nella voce Parameters, inserire gli NtpServer preferiti e specificare NTP su `Type`. Infine, sotto la voce `TimeProviders`, cercate `NtpServer` e impostate a `1` la chiave `Enabled`.
L'ultimo step è quello di riavviare il servizio, e lo facciamo con

```
C:\Windows\System32> net stop w32time && net start w32time
```

Se siete in un server di dominio, per segnalare ai client l'esistenza del server NTP dovrete utilizzare l'Editor dei Criteri di Gruppo:

- È sostanzialmente un programma che definisce le policy (regole) per la gestione di un Dominio Windows.
- Un Dominio è un insieme di Workstation e Utenti che operano all'interno di uno stesso ambiente condividendo risorse gestite attraverso regole e permessi (policy).

In generale (fuori da un dominio) anche per impostare un client Windows all'uso del server NTP bisogna agire sui registri di sistema (del client ovviamente).

DHCP (Dynamic Host Configuration Protocol): permette ad un server di assegnare dinamicamente IP, hostname, una lista dei DNS ed altre eventuali informazioni aggiuntive a degli host che ne fanno richiesta.

È inoltre possibile fare in modo di assegnare ad uno stesso host (o meglio, ad una stessa scheda di rete con il medesimo MAC address) sempre il medesimo IP. Con lo stesso meccanismo si possono assegnare indirizzi IP solo a MAC address conosciuti.

Come funziona? Semplicemente il client all'avvio della configurazione di rete invia in broadcast un pacchetto che ricerca un eventuale server dhcp.
Se è presente un server dhcp nella rete, esso riceve la richiesta, controlla eventualmente il mac address della scheda di rete che ha inviato la richiesta, e invia la configurazione di rete al client.
L'implementazione di dhcp più usata nei sistemi derivati da UNIX è ISC DHCP.
In debian/ubuntu si installa con I comandi:

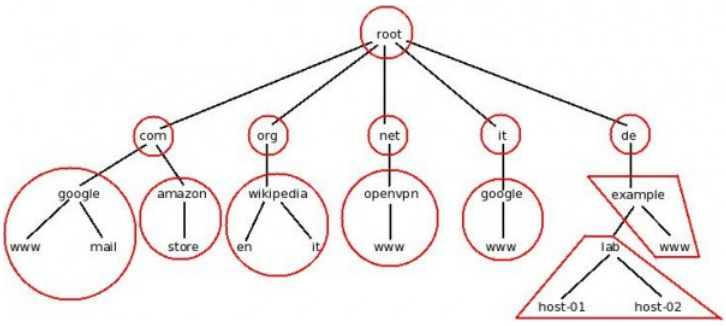
```
# aptget install iscdhcpserver
# aptget install iscdhcpclient
```

Per la configurazione del server, riferirsi al file in `/etc/dhcp.conf`.

Domain Name Server - DNS. Ad ogni indirizzo IP può essere assegnato un nome simbolico (es: `gundam.dsi.unive.it`).
Per far questo è necessario gestire in qualche modo una tabella che indica la corrispondenza tra nome e ip (diretta), ip e nome (inversa).
Questo è il compito del Domain Name System che viene gestito dai Domain Name Server.
Quindi un DNS è sostanzialmente un database distribuito, basato su modello C/S che traduce il nome di una macchina in un indirizzo internet e permette di:

- Amministrare la relazione tra nomi ed indirizzi del proprio dominio in maniera indipendente.
- Risolvere i nomi esterni al proprio dominio accedendo alle informazioni gestite da altre organizzazioni.

Possiede una duplice gerarchia: dei nomi di dominio e dei server che ne forniscono i dati relativi.
Per i nomi di dominio: posizione più alta occupata dai Top Level Domain: `com, it, edu ecc`. Al di sotto di essi vi altri domini: `unive.it, dais.unive.it` ecc..
Lato server le informazioni sono suddivise in zone. Zona ¡¿ sottodominio/dominio.
La gerarchia dei name server parte dai root name server e si realizza tramite deleghe sulle zone, aventi come radice la zona ".".



I DNS, oltre alla risoluzione da nomi a IP, provvedono anche alla risoluzione inversa. Quest'ultima avviene tramite la gerarchia di zone `in.addr.arpa` (`ip6.arpa` per IPv6).
Le zone contengono diversi tipi di informazioni, memorizzate in record, tra cui ricordiamo:

- SOA, che fornisce le informazioni generali sulla zona, tra cui un identificativo sequenziale della versione della definizione (serial number), l'indirizzo email del responsabile e il tempo di vita (TTL) della definizione della zona.
- NS, che definisce quali siano i DNS responsabili per le richieste iterative riguardanti la zona.
- MX, che definisce quali siano gli hostname dei server di posta in entrata responsabili per il dominio.
- A, che rappresenta traduzioni nella forma da nome a ip. Per IPV6 il tipo corrispondente è AAAA.
- CNAME, che rappresenta dei nomi alternativi (alias) per un host, nella forma da alias a nome.
- TXT, che permette di associare stringhe arbitrarie ad un nome, nella forma da nome a stringa.
- SRV, che permette di associare delle informazioni strutturate al nome di un servizio, nella forma da nomeservizio.protocollo a priorità peso porta nome host.
- Il tipo PTR che, nelle zone per la risoluzione inversa, rappresenta le traduzioni nella forma da ip a nome.

I DNS possono essere gestiti in due modi:

- Tramite l'elenco contenuto nel file `/etc/hosts`
- Tramite l'uso di un server DNS, in Linux è **BIND**

Alcuni comandi utili per la gestione dei DNS sono `host` e `dig`.
In Windows, oltre al sistema DNS, troviamo anche:

- WINS (Windows Internet Naming Service)
- NetBEUI (NetBIOS extended User Interface)

- NetBIOS (Network Basic Input Output System): avete presente quando nella rete di casa vostra sul file explorer scrivete \\`pc`\`cartella`? Quello

Tra i comandi utili, ricordiamo `netsh` e `nslookup`.
Nella stessa zona posso esserci più server dns, di solito riconosciuti come master (primario) contiene la copia principale del dabase dns e propaga agli altri le modfiche, slave (secondario) che possiede una replica, e caching.
Su Linux BIND, i files di configurazione li trovate in `/etc/bind`.

Capitolo 13

Autenticazione

Definizione. Si intende una serie di problemi e operazioni diverse che vanno dall'identificazione e autenticazione (chi sei) alle autorizzazioni vere e proprie (cosa vuoi e cosa puoi fare).
Storicamente, i sistemi UNIX usavano il contenuto `/etc/passwd` per entrambi gli scopi. Il file contiene tante righe quanti sono gli utenti del sistema, ed ognuna di queste ha la forma:

```
username:encpassword:UID:GID:gecos:homedirectory:shell
gsportel:xZELC6ObSje96:1959:1000:Giorgia Sportelli,,,:
/home/gsportel:/bin/sh
```

Poiché alcune delle informazioni contenute nel file (ad esempio la corrispondenza da username a UID) devono essere disponibili a tutti gli utenti di sistema (ed in particolare alle loro applicazioni), il file era (ed è) pubblicamente accessibile in lettura. La password era protetta dalla (supposta) robustezza dell'algoritmo di hashing usato, originariamente basato sull'algoritmo crittografico DES.
l file `/etc/group` contiene la lista dei gruppi del sistema e dei relativi GID, nonché, per ognuno di essi, l'elenco dei suoi membri, ad esclusione di quelli che hanno il suo GID come loro gruppo primario. La sintassi di ogni riga del file:

```
groupname:encpassword:GID:listamembri
```

Gli UNIX moderni non usano più direttamente il file `/etc/passwd` per la funzionalità che gli da il nome, cioè la memorizzazione della password. Questo ruolo è stato assunto dal file `/etc/shadow`, che, servendo solo per la fase di autenticazione, non è pubblicamente leggibile.

```
username:encpassword:ultimocambio:etàminima:etàmassima:
periodoavviso:periodoinattività:datascadenza:altro
```

NSS e PAM. Originariamente, tutti i file sopra indicati erano gestiti direttamente dai programmi che ne richiedevano la consultazione o la manipolazione (esempio vipw).
Ad ogni modifica dello standard del file era necessario modificare anche i programmi che gestivano gli stessi.
Allo stesso modo, sistemi che storicamente permettevano di centralizzare la gestione degli utenti su più sistemi, come NIS, richiedevano una nuova versione delle stesse utility.
Per ovviare a questa situazione, nel tempo sono nati sistemi che permettono,

tramite l'uso di componenti modulari, di aggiungere meccanismi di autenticazione e di gestione delle informazioni sugli utenti diversi, senza per questo dover modificare i programmi che ne fanno uso. In particolare, generalmente si usano:

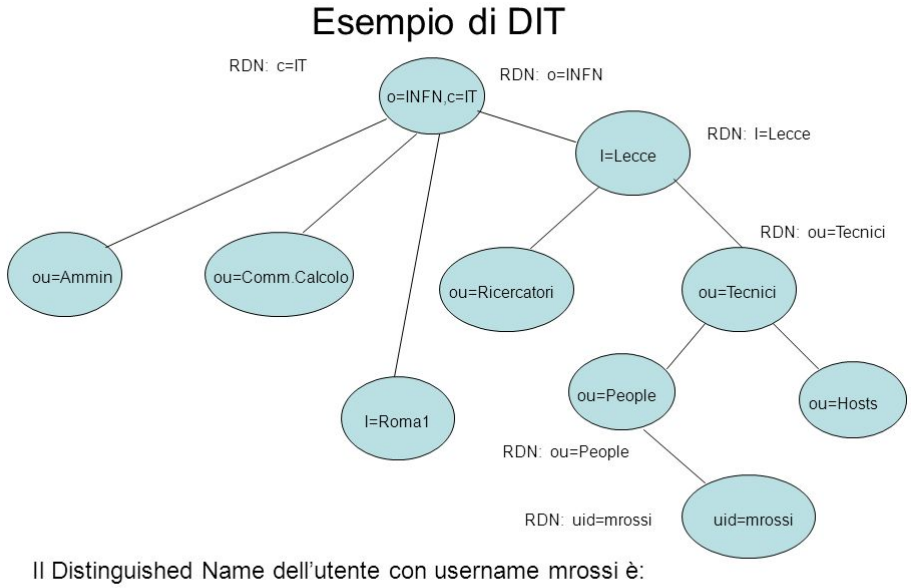
- nss (Name Service Switch) per le informazioni sugli utenti, nonché per altre mappe che associano nomi ad altre informazioni.
- PAM (Pluggable Authentication Modules) per i sistemi di autenticazione.

Il sistema nss, in Linux, è parte dell'implementazione delle librerie C del progetto GNU. Lo scopo del sistema è quello di fornire i dati dei system database (tra cui `passwd` e `group`) alle applicazioni. Le sorgenti dei dati, oltre ai file omonimi, possono essere le più disparate (ad esempio mysql, ldap, ecc).
Dei moduli, anche di terze parti, si occupano di tradurre richieste e risposte a/da questi database affinché possano essere effettuate sulla fonte prescelta, come ad esempio una directory LDAP o un database SQL. La configurazione di nss avviene principalmente tramite il file `/etc/nsswitch.conf`, mentre moduli aggiuntivi si trovano, solitamente, nei pacchetti Debian/Ubuntu dal nome `libnss-*`.
`nscd` è un daemon della cache di rete, ed è fondamentale per le performance in caso di autenticazione via rete.
Il sistema PAM fornisce i servizi d'autenticazione ad una o più applicazioni/servizi (services), che possono avere configurazioni separate. Le attività (task) di autenticazione stessa sono, in realtà, suddivise in 4 management group:

- **account:** si occupa di verificare eventuali requisiti che l'account deve rispettare, ad es. se è scaduto o se ha il permesso o divieto di accedere.
- **auth(entication):** si occupa di autenticare l'utente e di acquisire le credenziali necessarie.
- **password:** si occupa di aggiornare/modificare le informazioni di autenticazione. Un classico esempio è proprio l'operazione di cambio password.
- **session:** si occupa di gestire le attività che devono essere effettuate in apertura ed in chiusura di una sessione, ad esempio
  - all'accesso aggiorna I file `utmp` e `wmtp`, fa partire una sessione `sshagent`... item all'uscita aggiorna I file `utmp` e `wmtp` termina la sessione `sshagent`...

Per ogni servizio ed ognuna di queste attività si possono specificare una sequenza di operazioni tramite l'uso di moduli (o plugin) intercambiabili.
Vi sono molti plugin di terze parti, che in Debian/Ubuntu sono forniti in pacchetti dal nome `libpam-*`.
La configurazione si trova nei file nella directory `/etc/pam.d/`, oppure, anche se sconsigliato, nel file `/etc/pam.conf`.
I file in `/etc/pam.d/common-*` indicano configurazioni di default, che devono essere importate nei file che specificano eventuali modifiche del comportamento standard.

La strada verso LDAP e KERBEROS, Active Directory e Autenticazione Centralizzata. In principio esisteva NIS, che sta per Network Information Services, fu sviluppato da Sun Microsystems per centralizzare l'amministrazione di sistemi UNIX (in origine SunOS). Ora in sostanza è diventato uno standard di settore; tutti i sistemi UNIX like (Solaris, HP-UX, AIX, Linux, NetBSD, OpenBSD, FreeBSD, etc) supportano NIS.
NIS in precedenza era noto come Yellow Pages, ma per una questione di marchi, Sun ha cambiato il nome. Il vecchio termine (e yp) è ancora si incontra ancora spesso.
E' un sistema client/server basato su RPC che permette ad un gruppo di macchine in un dominio NIS di condividere un insieme comune di file di configurazione. Questo permette ad un amministratore di sistema di installare sistemi client NIS con il minimo di dati di configurazione e di aggiungere, rimuovere o modificare dati di configurazione da una singola macchina.
Questo sistema era molto insicuro, in quanto tutto viaggiava in chiaro. Bastava diventare root di un client per accedere a tutte le informazioni del database NIS.
Poi arrivò NIS+, con una struttura gerarchica con server principali e di backup, e autenticazione tramite password criptata. Ora non è più supportato.
Tornando ai giorni nostri: LDAP (Lightweight Directory Access Protocol) è un protocollo per l'interrogazione e la manipolazione di directory, ovvero basi di dati che seguono un modello gerarchico.
Viene utilizzato in ambiti in cui, il numero delle operazioni di consultazione (lettura) può essere elevato, mentre le operazioni di modifica (scrittura) sono relativamente rare.
DIT (Directory Information Tree): è l'albero dei dati contenuti in un database ldap.
I nodi rappresentano suddivisioni di vario tipo (ad esempio gli uffici di una organizzazione). Le foglie sono i dati veri e propri.
Ogni nodo (entry) dell'albero ha un relative distinguished name (RDN), che è unico tra tutti i nodi fratelli. Mentre la sequenza di tutti gli RDN dalla radice fino al nodo interessato viene chiamato Distinguished Name (DN), ed è unico per tutto il DIT.
I nodi possono avere informazioni strutturate (ad esempio nome, cognome, email, numero di telefono...) organizzate in attributi, ed implementare uno o più schemi (ing. schema), che non sono altro che delle definizioni di tipi di dato.
I nodi possono anche indicare dei riferimenti ad altre parti del DIT, o addirittura a DIT di altri server LDAP (referral).

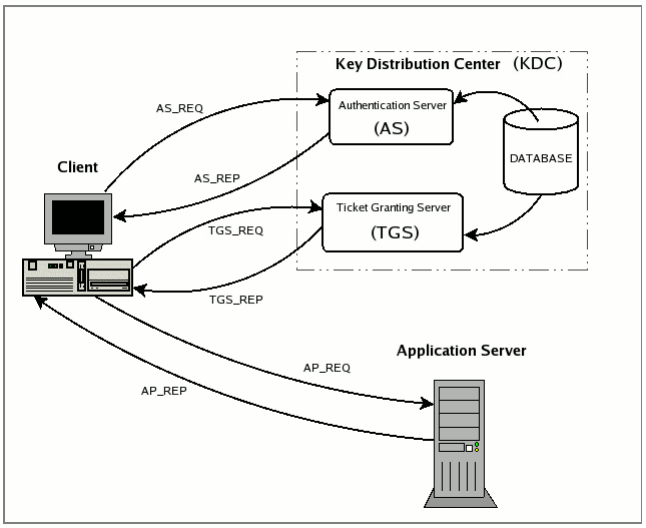


L'accesso alla directory ldap può essere anonimo oppure tramite autenticazione (binding). La fonte dei dati d'autenticazione può essere interna al DIT o esterna.
Vi sono molte implementazioni di server LDAP. Le più note nei sistemi UNIX sono OpenLDAP (pacchetto Debian/Ubuntu slapd) e 389 Directory Server.
Su Windows si utilizza Active Directory (con Kerberos).

LDAP è spesso usato per fornire le informazioni sugli utenti e sui gruppi ad un insieme di elaboratori. Il sistema nss dispone di un plugin (pacchetto Debian/Ubuntu `libnss-ldap`) che, attivato e configurato opportunamente, consente di usare dei server ldap come fonte per i vari database di sistema.
LDAP, qualche volta, è anche usato (tramite `libpam-ldap`) per l'autenticazione, facendo verificare al sistema locale se la password cifrata presente nel DIT (esattamente come in `/etc/passwd` o `/etc/shadow`) corrisponde a quella immessa dall'utente, oppure cercando di autenticarsi (binding) al server LDAP stesso usando quelle credenziali.
Kerberos è un sistema per l'autenticazione distribuita basato sulla crittografia. Il protocollo si affida ad un insieme limitato di nodi fidati che fanno da arbitro alla situazione richiedendo e fornendo prove dell'avvenuta identificazione ed autenticazione delle parti.
I sistemi che usano una stessa infrastruttura basata su Kerberos formano un Realm, che ha un nome ben definito.

Schema di funzionamento di Kerberos:

- AS_REQ è la richiesta iniziale di autenticazione dell'utente (fatta con il kinit tanto per intenderci). Tale messaggio è diretto alla componente del KDC nota come Authentication Server (AS);
- AS_REP è la risposta dell'Authentication Server alla richiesta precedente. Sostanzialmente contiene il TGT (criptato con la chiave segreta del TGS) e la chiave di sessione (criptata con la chiave segreta dell'utente richiedente);
- TGS_REQ è la richiesta da parte del client rivolta al Ticket Granting Server (TGS) per un ticket di servizio. Dentro questo pacchetto viaggia il TGT ottenuto dal messaggio precedente e un autenticatore generato dal client e criptato con la session key;
- TGS_REP è la risposta del Ticket Granting Server alla richiesta precedente. Ci si trova dentro il service ticket richiesto (criptato con la chiave segreta del servizio) e una chiave di sessione di servizio generata dal TGS e criptata con la precedente chiave di sessione generata dall'AS;
- AP_REQ è la richiesta che il client manda ad un server applicativo per accedere ad un servizio. Le componenti sono il ticket di servizio ottenuto dal TGS con la risposta precedente e un autenticatore generato sempre dal client, ma questa volta criptato con la chiave di sessione del servizio (generata dal TGS);
- AP_REP è la risposta che il server applicativo dà al client per provare di essere veramente il server che il client si aspetta. Questo pacchetto non è sempre richiesto. Il client lo richiede al server solo quando è necessaria la mutua autenticazione.



Esistono diverse implementazioni di Kerberos. In ambiente UNIX le più note sono MIT Kerberos e Heimdal. Mentre il protocollo Kerberos è standard, e pertanto le implementazioni sono interoperabili, alcune funzionalità di amministrazione, che usano un protocollo ad hoc, possono non esserlo.
Tramite il modulo PAM appropriato (pacchetto `libpam-krb5`) è possibile far

autenticare un utente del sistema usando le sue credenziali Kerberos. Il modulo si occuperà anche di far avere un TGT all'utente, che così potrà usare altri servizi che partecipano al realm senza dover fornire nuovamente la password.
Sebbene anche i servizi possano usare Kerberos attraverso PAM, questo non permette la funzionalità di single sign-on appena descritta, né altre caratteristiche avanzate di Kerberos. Pertanto i servizi che lo richiedono spesso usano Kerberos per conto loro o attraverso GSSAPI (Generic Security Service Application Program Interface).
Active Directory è un database ldap integrato nei server Windows ($\geq$ 2000), che fungono da Domain Controller, e consente di catalogare e gestire in modo centralizzato risorse di vario genere come: utenti, gruppi di lavoro, stampanti, cartelle condivise, ecc.
La struttura del database è di tipo gerarchico, con contenitori che contengono oggetti e altri contenitori.
Sostanzialmente è una implementazione limitata di ldap + kerberos.
Il primo procedimento da attuare e definire la struttura di Active Directory. Active Directory è un contenitore di oggetti, ovvero utenti, gruppi, computer, server, stampanti, unità organizzative.
Per la realizzazione di un dominio active directory è sufficiente lanciare il comando `dcpromo.exe`. Il sistema si occuperà da solo di creare tutto ciò che serve per l'autenticazione centralizzata, tra cui anche un server DNS.
Importanti per un corretto funzionamento di ldap, kerberos e active directory sono:

- Sincronizzazione degli orologi (NTP).
- Corretta configurazione dei DNS sia diretta che inversa.
- Correttezza dei certificati SSL.
- Manutenzione costante del database utenti/gruppi.

Capitolo 14

Posta

La posta è uno dei servizi più diffusi e importanti della rete Internet.

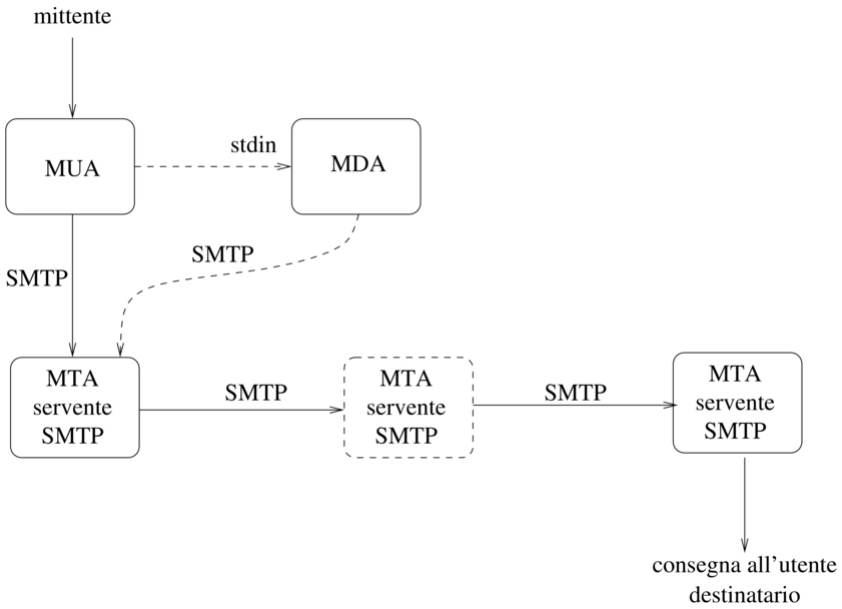
Struttura Il servizio di posta è così strutturato:

- **Mail User Agent:** client di posta, ovvero un programma usato da un utente per inviare/visualizzare messaggi, può anche essere una webapp.
- **Mail Submission Agent:** si occupa di ricevere i messaggi da un MUA e inviarli ad un MTA.
- **Mail Transit Agent:** si occupa di ricevere mail da un MSA o da un altro MTA, e a instradarle a un altro MTA oppure a un LDA.
- **Mail Delivery Agent o Local Delivery Agent:** si occupa di consegnare il messaggio alla casella di posta dell'utente indicato qualora la destinazione finale del messaggio sia nel sistema corrente.
- **Mail Access Agent:** permett di visualizzare/scaricare i messaggi.
- **Mail Retrival Agent:** scarica la posta da un MAA e la rende disponibile in locale.

Va tenuto conto che MSA è integrato nel MTA, e MRA è integrato nel MUA.
Il servizio di Posta segue un certo flusso:

1. Un utente (mittente) scrive una email usando un MUA.
2. MUA invia la mail ad un MSA/MTA.
3. MTA controlla l'indirizzo di destinazione (`utente@dominio`):
    - Se il dominio è tra quelli serviti da MTA in questione (indirizzo locale) e utente è effettivamente valido, questa viene girata al LDA, che la consegna nella casella di posta associata, e il viaggio termina; altrimenti, MTA rifiuta il messaggio.
    - Se invece l'indirizzo non è locale, MTA accetta di instradare il messaggio (relay), mette il messaggio in una coda d'uscita e procede.

4. Dopo aver estratto dalla coda il messaggio, il MTA controlla quale sia il record MX associato al dominio o, se non presente, un record 'A' (relativo al nome DNS dell'host dato), e contatta l'MTA che risponde a quell'host, cercando di inviargli il messaggio.
    - Se l'invio avviene correttamente, il messaggio è gestito da MTA di destinazione, che procede dal punto `2)`.
    - Se MTA contattato non risponde, il messaggio torna in coda.
    - Se MTA contattato rifiuta il messaggio, oppure se il messaggio è stato troppo tempo in coda, viene mandata una mail all'indirizzo indicato dal mittente, notificando la mancata consegna, e il procedimento termina.
5. A questo punto il messaggio si trova nella `inbox` del destinatario.



Il flusso, così com'è, non garantisce in alcun modo la consegna, la notifica degli errori, l'identità del mittente o la privatezza della comunicazione.
Per ovviare a questi problemi, bisogna ricorrere a sistemi a livello applicazione.
Il relay non è sempre concesso. Si accetta in due casi:

- Se la richiesta viene da un host conosciuto (es.: la propria rete aziendale).
- Se l'utente è autenticato.

In tutti gli altri casi il messaggio viene rifiutato immediatamente.

Protocolli Il protocollo usato tra MUA e MTA e tra MTA e MTA si chiama **SMTP** (Simple Mail Transfer Protocol) o ESMTP (Extended SMTP). La comunicazione tra MTA e LDA può avvenire sia internamente (es.: tramite scambio di file e/o memoria condivisa tra le componenti), oppure tramite **LMPT**

(Local Mail Transfer Protocol), che è una versione semplificata di ESMTP. La connessione tra MAA e MUA avviene tipicamente tramite **POP3** (Post Office Protocol versione 3) o **IMAP4** (Internet Message Access Protocol versione 4).

Differenze tra POP e IMAP:

- **IMAP** conserva le mail sul server; la lettura, l'invio e la gestione possono avvenire anche da client desktop, ma è il server a mantenere la copia delle mail inviate, ricevute e scritte; quindi con IMAP le mail sono disponibili da qualsiasi dispositivo.
- **POP** invece delega al dispositivo usato per la consultazione il compito di provvedere al salvataggio; le mail vengono scaricate sul dispositivo e la connessione è necessaria solo per inviare e ricevere posta; in realtà, i MUA hanno delle opzioni per POP che permettono di mantenere una copia dei messaggi sul server.

Sendmail fu il primo MTA a fare uso di SMTP. Venne utilizzato fino al 2005 come MTA da moltissimi server di posta, ma purtroppo la progettazione rigida e la complessità della configurazione lo resero via via meno popolare, fino all'estinzione. Un altro aspetto da considerare, era la presenza di numerosi bug e problemi di sicurezza.
Altri MTA degni di nota sono:

- Qmail: ideato da Dan Bernstein circa 10 anni fa, è un server di posta scalabile, performante, sicuro e portabile. Si dice sia il più sicuro... Attualmente è il secondo MTA più usato su internet. (Per approfondire: http://www.qmail-ldap.info).
- Courier MTA: è un server di posta/groupware integrato che poggia su vari protocolli: ESMTP, IMAP, POP3, SSL e HTTP. Sostanzialmente offre tutti i servizi di posta elettronica compreso anche un sistema di webmail. È quindi una soluzione completa per la gestione della posta, volendo è simile a Exchange. (http:// www.courier-mta.org)
- Exim: MTA standard di Debian fino a qualche versione fa.
- Zimbra: suite completa per la gestione della posta (http://zimbra.org).
- Postfix.
- Exchange.

	Exim	Postfix	Sendmail	Qmail
Sicurezza	Medio-bassa	Alta	Bassa	Molto alta
Installazione	Media	Medio-facile	Facile	Medio-difficile
Configurazione	Medio-facile	Facile	Difficile	Facile
Performance	Medie	Alte	Medie	Alte
Maturità	Bassa	Media	Alta	Media
Documentazione	Molta	Media-molta	Molta	Molta
Features	Medie-molte	Medie	Molte	Poche*
* disponibili patch per nuove funzionalità, necessitano ricompilazione				

Postfix è il MTA di riferimento in ambiente Linux; è facile da configurare, è modulare, e permette di utilizzare diversi tipi di autenticazione. Permette inoltre di interfacciarsi facilmente con altri sistemi di controllo della posta:

- Real-time Blackhole List (RBL): viene bloccata una lista di indirizzi IP che sono ritenuti responsabili dell'invio di spam.
- Sender Policy Framework (SPF): sistema di validazione mail progettato per riconoscere le mail provenienti da un indirizzo contraffatto (`spoofing`) che funziona grazie ad un meccanismo che permette al MTA di vedere se una mail arriva da un dominio autorizzato da chi amministra l'host.
- Sistemi di greylisting: metodo per difendersi dallo spam che consiste nel rifiutare temporaneamente tutte le mail da un mittente sconosciuto, in quanto se la mail è legittima il mailserver del mittente ritenterà l'invio; questo discorso era valido finché gli spammer non hanno capito che bastava rinviare le mail per bypassare la greylist.
- Sistemi antispam basati sul contenuto (i.e.: SpamAssassin): sistemi che filtrano le mail basandosi sul contenuto della mail.
- Sistemi antivirus e molto altro (i.e.: ClamAV).

MS Exchange è un software studiato per agevolare la collaborazione in linea tra vari utenti. Venne introdotto nel 1996 da Microsoft e oggi è uno dei mailserver più potenti e utilizzati. Tra le sue funzionalità più rilevanti, la gestione centralizzata della posta elettronica, dei calendari e delle rubriche contatti, che possono essere condivisi tra i vari utenti di una rete aziendale. Il client più utilizzato per connettersi a un server Exchange è MS Outlook, disponibile nella suite Office. È disponibile anche una webapp (Outlook Web Access) per l'accesso da browser.

MAA Dovecot permette di usare i protocolli IMAP e POP3 e supporta, sia in consultazione che in consegna, diversi formati di memorizzazione della posta,

quali `mbox, maildir, dbox`. Permette inoltre, tramite un plugin, di avere un sistema di filtri server side con cui smistare la posta degli utenti in vari folder, nonché poterla inoltrare ad altri indirizzi. Per gestire questi filtri lato server si usa il protocollo SIEVE. Consente anche di gestire la quota della posta per ogni utente, di adottare vari metodi di autenticazione tramite SQL, LDAP, ecc..

Spam: uno o più messaggi non richiesti, inviati o postati come parte di un più grande insieme di messaggi, tutti con il medesimo contenuto.
Possiamo dividere lo spam in cinque categorie:

- **Hoax:** bufale o catene di messaggi.
- **Worm:** mail mandate da un virus.
- **UCE:** Unsoliticed Commercial Email, ovvero mail di carattere commerciale non richieste.
- **UBE:** Unsoliticed Bulk Email, ovvero mail indesiderate mandate in grande quantità.
- Messaggi derivanti da mailing list.

Le maggiori fonti di spam sono UCE e UBE. La prevenzione e la lotta allo spam non sempre sono facili, in quanto vengono sempre creati nuovi metodi per far pervenire le mail indesiderate alle nostre caselle di posta.
Appare evidente che il problema dello spam non è arginabile da una policy statica, per quanto possa essere complessa. Possiamo solo decidere di filtrare le mail in ingresso con vari meccanismi di filtraggio, ma tenendo conto che non deve essere totale in quanto si rischierebbe di compromettere il servizio mail scartando delle mail legittime.

Greylisting: tecnica di difesa dallo spam che consiste nel rifiutare la mail proveniente da un dominio sconosciuto per un numero `N` di volte. Ad ogni rifiuto, MTA sender rimette la mail in coda e dopo un certo periodo di tempo `T` ritenta l'invio. Al tentativo `N`, la mail viene accettata e MTA di invio viene inserito in una whitelist, in modo da considerare legittime le mail provenienti da questo.
Il greylisting è una tecnica che sta perdendo efficacia, in quanto i programmi che si occupano dell'invio dello spam sono ingrado di reinviare più volte la mail, aggirando il sistema; inoltre, possono verificarsi ritardi nella consegna delle mail.
L'aspetto positivo è che con questo sistema, nessuna mail lecita viene scartata.
In postfix, il greylisting è implementato grazie ai demoni `postgrey` o `policyd`.

RBL: sono "liste nere" contenenti un elenco di indirizzi IP che non sono autorizzati ad inviare email.
Non è facile distinguere gli indirizzi IP validi da quelli di spam, e spesso vengono scartate delle mail lecite. Spesso quindi vengono implementati dei criteri di utilizzo "personali", e vengono gestite da terze parti.
Alcuni criteri per la selezione degli IP da bloccare includono il blocco di tutti

gli IP assegnati dinamicamente dal provider, il blocco di tutti gli IP che inviano mail senza passare da un mail server ufficiale e il blocco di IP segnalati come spammer dagli utenti.
In postfix si attiva RBL alla direttiva `reject_rbl_client` alla voce `smtpd_recipient_restrictions`:

```
reject_rbl_client zen.spamhaus.org
```

SPF: non si tratta di una vera e propria tecnica di difesa dallo spam, ma piuttosto di uno standard che si applica in ambito di risoluzione dei DNS per cui si può dichiarare, tramite un record di testo, quali sono gli indirizzi IP o nomi che possono inviare mail per il dominio stesso.
Si crea sostanzialmente una maschera per cui il mailserver ricevente, se il record ha la giusta formattazione, può verificare se il server mittente è abilitato a inviare.
Questo comporta non poche problematiche:

- Sono ancora molti i domini che non implementano il suddetto record.
- L'implementazione può essere difficoltosa sulle strutture mail di una certa complessità.
- Si complica il processo di forwarding (inoltro) delle mail.

In postfix ci si rifà alla direttiva `smtpd_recipient_restrictions`, inserendo la voce:

```
check_policy_service spf
```

Spamassasin: è una tecnica di difesa dallo spam che utilizza un sistema di filtri euristici, ovvero il sistema prova a "indovinare" se la mail è valida o meno assegnandole un punteggio sulla base di vari aspetti, come per esempio la lingua della mail, la presenza di tag `html` o la presenza di parole chiave.
A seconda del punteggio ottenuto dalla mail, spamassasin può decidere di lasciar passare la mail, applicare dei tag all'oggetto della mail per segnalarla all'utente come spam, oppure scartarla.
Il vantaggio è che può essere "istruito" in base alle mail ricevute in precedenza, e di conseguenza migliorare la sua efficacia se si verificano determinate condizioni nelle mail ricevute in precedenza.
Per essere correttamente configurato necessita di continui aggiustamenti fatte da personale con una certa esperienza.
In postfix si presenta con il daemon `spamassassin`.

Controlli MTA: MTA può implementare dei controlli per ridurre il flusso dello spam automatizzato.
Ad esempio, in postfix:

- `smtpd_helo_required = yes` controlla che il sender inizi la comunicazione con il comando `ehlo`.
- `smtpd_helo_restrictions` controlla chi può iniziare un dialogo tramite il comando `ehlo`. I parametri sono:
  - `permit_sasl_authenticated`
  - `permit_mynetworks`

  - `reject_non_fqdn_hostname`
  - `permit`
  - ...

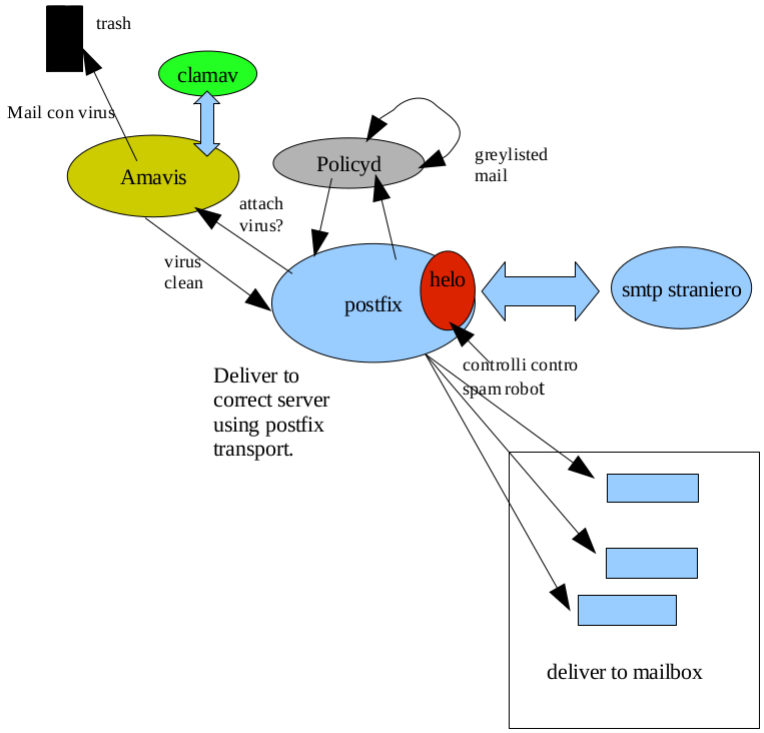
- `smtpd_sender_restrictions` controlla chi possa inviare mail dopo essersi identificato con `ehlo`. I parametri sono:
  - `permit_sasl_authenticated`
  - `permit_mynetworks`
  - `reject_non_fqdn_sender`
  - `permit`
  - ...

- `smtpd_recipient_restrictions` effettua controlli sul destinatario. I parametri sono:
  - `check_policy_service`
  - `inet:127.0.0.1:10031`
  - `reject_non_fqdn_recipient`
  - `permit`
  - ...

AMAVIS: è un sistema che filtra i contenuti delle mail, implementando il trasferimento, l'elaborazione e la codifica delle stesse, interfacciandosi con altri sistemi di filtraggio di spam e virus.
Può essere visto come un'interfaccia tra un MTA e altri sistemi di filtraggio, come spamassassin.
Può essere usato per rilevare virus, spam, contenuti vietati nella mail, etichettatura, blocco e spostamento delle mail a seconda del loro contenuto, messa in quarantena o rilascio dei messaggi, eliminazione dei virus dai messaggi grazie ad antivirus esterni, ecc..
Una configurazione comune di Amavis è quello di avere un sistema di filtraggio basato su:

- Postfix
- Spamassassin
- ClamAV
- Amavis

In postfix è presente come daemon `amavisd-new`.



Per installare postfix si utilizza il comando:

```
#apt-get install postfix
```

I files di configurazione sono presenti in `/etc/postfix`.
per installare dovecot si utilizza il comando:

```
#apt-get install dovecot dovecot-imapd dovecot-pop3d dovecot-lmtpd
```

I files di configurazione in `/etc/dovecot`.

```
#
==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (no)    (never) (100)
#
==========================================================================
smtp      inet  n       -       y       -       -       smtpd
smtps     inet  n       -       y       -       -       smtpd
pickup    unix  n       -       y       60      1       pickup
cleanup   unix  n       -       y       -       0       cleanup
qmgr      unix  n       -       n       300     1       qmgr
```

```
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache


smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated
defer_unauth_destination
myhostname = gundam.dsi.unive.it
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = $myhostname, gundam, localhost.localdomain, , localhost
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
 mail_location = mbox:~/mail:INBOX=/var/mail/%u
 namespace inbox {
   inbox = yes
   location =
   mailbox Drafts {
     special_use = \Drafts
   }
   mailbox Junk {
     special_use = \Junk
   }
   mailbox Sent {
     special_use = \Sent
   }
   mailbox "Sent Messages" {
     special_use = \Sent
   }
   mailbox Trash {
     special_use = \Trash
   }
   prefix =
 }
 passdb {
   driver = pam
 }
 protocols = " imap pop3"
 ssl = no
 userdb {
   driver = passwd
 }
```

Capitolo 15

Filesystem Distribuito

Il filesystem distribuito è un particolare tipo di filesystem che permette la memorizzazione dei files e delle risorse in dispositivi di archiviazione distribuiti in una rete informatica.
I dati non sono archiviati su un dispositivo locale ma, attraverso un sistema client/server, su dispositivi remoti collegati in modo trasparente alla propria gerarchia dei file.
Un DFS deve gestire i files in modo concorrente e trasparente, e può essere dotato di autenticazione e criptazione.

File Server: è un server che ospita un DFS offrendo una serie di servizi ai client che lo sfruttano. Sui client è installata un'interfaccia al file server che include delle operazioni che normalmente si fanno sui files locali.
Su NFS e CIFS questa interfaccia è trasparente: si utilizzano i comandi di sistema sui files come se fossero locali.
Il file server controlla un insieme di dispositivi di memoria di massa su cui agisce in base alle richieste dei client.
Un file server, per regola generale, deve essere replicato e ridondato.
In un filesystem distribuito i dispositivi di memorizzazione sono dislocati in una rete. Le richieste e le risposte devono, pertanto, essere trasportate attraverso tale rete. La differenza principale sta nel fatto che invece di avere un dispositivo unico e centralizzato, il sistema ne può avere molti e indipendenti.
Le implementazioni di un filesystem distribuito possono variare:

- In alcune occasioni il server (file server) viene eseguito su una macchina dedicata.
- Il sistema client può accedere simultaneamente a più file servers.
- La stessa macchina può ospitare sia un server che un client.

Idealmente un FS distribuito appare all'utente come un normale FS centralizzato, in quanto molteplicità e dispersione di server e dispositivi a cui si riferisce possono essere celati. L'interfaccia rivolta alle applicazioni che usa l'utente non dovrebbe poter distinguere i file locali da quelli remoti, in quanto è compito del FS distribuito trovare e trasportare i dati attraverso la rete. L'aspetto che

incide maggiormente sul lato utente sono le prestazioni del FS distribuito.
Per valutare le prestazioni di un FS distribuito si quantifica l'ammontare di tempo impiegato per soddisfare una data richiesta. Nei sistemi convenzionali questo consiste nell'accesso al disco locale e ad una piccola quantità di elaborazione da parte della CPU. Nei sistemi distribuiti, invece, si somma il ritardo dovuto alle comunicazioni di rete: questo ritardo include il tempo necessario a sottoporre la richiesta al server e quello per ottenere la risposta attraverso la rete, mentre occorre sommare il tempo necessario alla CPU per manipolare il protocollo di comunicazione in ciascuna direzione. Le prestazioni di un FS distribuito incidono sul suo livello di trasparenza, in quanto un sistema distribuito dovrebbe avere una velocità paragonabile a quella di un sistema convenzionale.
Un FS distribuito deve provvedere sia all'accesso dei client ai files, ma anche alla loro modifica: gli aggiornamenti operati da un client non possono però interferire con gli accessi e le modifiche fatte da altri client. Sono quindi necessari meccanismi di controllo della concorrenza e locking che possono essere inclusi nella realizzazione del FS distribuito, oppure resi disponibili da un protocollo parallelo.
Nel tempo sono stati sviluppati diversi DFS, in particolare:

- **NFS:** il primo DFS a essere sviluppato (dalla SUN nel 1985), diffuso e molto usato, ora è alla versione 4.
- **AFS**
- **CIFS/SMB**
- **Google FS**
- **Coda, Plan9, xFS**

NFS: usa un remote access model, dove i nodi client non sanno realmente dove sono i files, e i server esportano una serie di operazioni sui files.
Altri sistemi usano un modello upload/download:

- Il nodo client scarica il file in una cash locale.
- A modifiche avvenute, il file viene caricato sul server.
- Il server dovrebbe mantenere le versioni dei files; tra i sistemi di versioning utilizzabili: SVN, GIT, CVS.

NFS è indipendente dall'organizzazione del FS locale, e per questo riesce a integrare i FS di Unix, Linux, Windows e OS X.
Esporta all'utente una visione simile a quella dei FS Unix-like basati su files organizzati come sequenze di bytes. Utilizza RPC come protocollo sottostante, ovvero un client fa una richiesta remota di esecuzione di una subroutine per la gestione dei files.
Dalla versione 4 supporta server non stateless (che richiedono, quindi, il ricevimento di un ACK seguente all'invio di un pacchetto). I client dovevano mantenere lo stato delle operazioni correnti su un FS remoto, ma nella versione 4 i server NFS mantengono lo stato delle operazioni. A partire dalla versione 4, inoltre, è stato introdotto il supporto del TCP, mentre prima era utilizzabile UDP.

Nella versione 4, NFS ha introdotto le compound procedure, che comprendono più richieste di operazioni in una singola chiamata, riducendo così il numero di chiamate RPC (offrendo, quindi, prestazioni migliori).
Riguardo le compound procedures, va tenuto conto che:

- Se un'operazione in una compound procedure fallisce, le successive operazioni non vengono eseguite.
- Viene ritornato un messaggio con le informazioni sulle operazioni eseguite e l'errore che si è verificato.
- Non conviene inviare operazioni non correlate tramite una compound procedure.

Quando si tratta di un FS wide area, è bene implementare file locking, protocolli di cache consistency e procedure ci callback.
Riassumendo:

- NFSv4 si basa sul protocollo TCP.
- Il server deve mantenere delle informazioni di stato.
- Il riconoscimento dell'utente avviene tramite una stringa arbitraria (es.: username); la traduzione di queste stringhe e le informazioni necessarie a cliente e server avviene tramite un `id mapper` (daemon `imapd`).
- L'identità degli utenti può essere provata tramite sistemi di autenticazione esterni (Kerberos).
- Vengono introdotte nel protocollo delle possibili migliorie delle prestazioni (es.: le deleghe) e per l'affidabilità (supporto per la replicazione, che però non è implementata direttamente nel protocollo).

Per installare NFS in ubuntu/debian:
`# apt-get install nfs-kernel-server` (sul server)
`# apt-get install nfs-common` (sul client)
Per la configurazione, editare `/etc/exports`, che ha la sintassi `/percorso cartellacondivisa indirizzoip (opzioni)`. Ecco alcuni esempi:

```
/home 157.138.22.21(rw, no_root_squash)
/home 157.138.0.0/16 (rw,async,no_subtree_check,sec=krb5:krb5i:krb5p)
```

Per la gestione del servizio, invece:

```
# service nfs-kernel server restart
```

```
# /etc/exports: the access control list for filesystems which may be
exported
# to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check)
hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
```

Per utilizzare l'area condivisa ho tre modi:

- Effettuando il mount manuale:
  `# sudo mount 192.168.1.1:/home /mnt/nfs/home`
- Inserendo il mount `/etc/fstab` in modo che sia montato in avvio:
  `192.168.1.1:/home /mnt/nfs/home nfs rw,user,auto 0 0`
- Usando **automount**: soluzione migliore in caso di molti client e molti utenti che non hanno accesso diretto all'utente amministratore (`root`) e/o ai files di sistema (`/etc/fstab`).

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to
name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>
<pass>
proc            /proc           proc    nodev,noexec,nosuid 0
  0
/dev/mapper/g-r /               ext4    errors=remount-ro 0
1
/dev/mapper/g-h /home           ext4    defaults    0       2
/dev/mapper/g-s none            swap    sw              0       0

daffynas:/volume1/data  /data  nfs vers=3          0    0
```

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name
devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
/dev/sda5 /               ext4    errors=remount-ro 0       1
/dev/sda6 none            swap    sw                0       0
roar.dsi.unive.it:/s1      /stud/s1    nfsvers3        0       0
roar.dsi.unive.it:/s2      /stud/s2    nfsvers3        0       0
roar.dsi.unive.it:/s3      /stud/s3    nfsvers3        0       0
roar.dsi.unive.it:/s4      /stud/s4    nfsvers3        0       0
```