

Bootstrap: codice d'avvio

- Abbiamo visto a inizio corso che il bootstrap del SO è effettuato da un programma residente in una piccola memoria speciale ROM o EEPROM
- questa memoria ha un indirizzo noto alla CPU, che viene consultato all'avvio del computer
- il fatto che si tratti di una ROM fa sì che il **codice d'avvio non sia modificabile**, per esempio, da virus: la modifica richiede una sostituzione fisica del circuito

Bootstrap: codice d'avvio

- Abbiamo visto a inizio corso che il bootstrap del SO è effettuato da un programma residente in una piccola memoria speciale di tipo ROM
- questa memoria ha un indirizzo noto alla CPU, che viene consultato all'avvio del computer
- il fatto che si tratti di una ROM fa sì che il codice d'avvio non sia modificabile, per esempio, da virus: la modifica richiede una sostituzione fisica del circuito
- un utente può però installare diversi SO, magari più d'uno sulla stessa macchina, come fa il codice di avvio (un programma molto semplice, installato prima di qualsiasi SO) a sapere quanti SO reperire il loro codice?
- molti produttori risolvono il problema inserendo nella ROM soltanto un programma, detto **boot loader**, che carica **il vero e proprio programma di avviamento, residente su disco e modificabile**

Disco di avvio

- il programma di avviamento risiede in alcuni blocchi di una partizione del disco detti blocchi di avviamento
- il disco contenente tale partizione è detto **disco di sistema** o disco di avvio
- il codice in ROM è solo un loader del programma di avviamento vero e proprio, che è a sua volta un loader del SO
- **Esempi**
 - NTLDR: **Windows NT Loader**
 - Linux Loader: **LILO**
 - **GRUB**
- **NB:** alcuni SO sono così strettamente legati all'HW su cui girano (es. Macintosh pre-1995) che è quasi impossibile far caricare sullo stesso HW un SO diverso

Boot-loader primario e secondario

- il **BIOS** è il boot-loader primario, oltre a caricare un boot-loader secondario, funzione del BIOS è eseguire il POST (power-on self-test, verifica che i device abbiano corrente) e inizializzare tutti i device che servono prima che il SO sia stato avviato, es:
 - monitor e tastiera
 - mouse,
 - controller SCSI, USB, CDROM,
 - ...
- la ricerca del boot-loader secondario viene fatta secondo la sequenza di device di memoria contenuta in una lista definita dall'utente
- **esempio**: se la sequenza è CDROM, HD cercherà prima se è presente un CD (Live CD) e se questo contiene un SO, se non lo trova passa a cercare sull'HD
- sull'HD la posizione del loader secondario è spesso indipendente dal SO installato. Si fa riferimento a una locazione nota come **MBR, master boot record**

Hard e soft reboot

- l'avvio di un computer dipende in parte dalla modalità di spegnimento attuata in precedenza. Si parla di **hard (o cold) reboot** e **soft (o warm) reboot**
- **Hard reboot**: il computer si è spento a causa di un'interruzione di corrente (o di un errore) oppure lo spegnimento forzoso è stato attuato deliberatamente per es. per contrastare l'azione di un virus
 - in questo caso il **contenuto di tutti i buffer e le cache viene perso**, il **FS può risultare in uno stato di inconsistenza** e occorre attuare procedure speciali di ripristino
- **Soft reboot**: segue uno spegnimento avvenuto attraverso una corretta procedura di shutdown

Boot-loader secondario

- GRUB, LILO, NTLDR sono esempi di boot-loader secondari
- **NT LoaDeR**::viene installato nell'MBR ed esegue queste operazioni nell'ordine:
 - accede al file system (di tipo FAT o NTFS)
 - verifica se esiste un'immagine dello stato di esecuzione relativa all'ultimo spegnimento (immagine di **ibernazione**), se sì carica l'immagine per consentire il proseguimento dell'esecuzione dal dove si era interrotta
 - se non la trova esegue il programma **boot.ini** e segue le scelte dell'utente
 - se è stato scelto un OS NT verifica l'HW presente e avvia il programma Ntoskml.exe che avvia NT
 - altrimenti cede il controllo al file associato alla scelta fatta nel file boot.ini, un file non editabile direttamente dall'utente
- **NB**: non è propriamente multiboot perché non consente di avviare SO sviluppati da altri produttori

Difetti fisici dei dischi

- nessun disco è completamente sano, quando escono dalla fabbrica i dischi presentano sempre **un certo numero di blocchi difettosi**
- altri possibili guasti possono riguardare le parti mobili (braccio, testine)
- **alcuni guasti sono irreparabili**: una testina che plana su un piatto causa un danno non recuperabile. L'unico modo per recuperare i dati è attraverso una **copia di backup** (es. delle home degli utenti)
- per i blocchi difettosi sono invece state studiate tecniche che consentono di usare un disco nonostante la loro presenza
- i blocchi difettosi (1) vanno individuati e (2) occorre “saltarli” quando si allocano blocchi ai file

Gestione manuale

- es. dischi con **interfaccia IDE**
 - i blocchi difettosi fin dall'origine vengono individuati dalla procedura di formattazione logica e opportunamente marcati come difettosi all'interno delle strutture usate per tener traccia dei blocchi liberi
 - es. format di MS-DOS segna i blocchi guasti nella FAT
 - i blocchi che si guastano nel tempo causano una perdita dei dati conservati in essi
 - esistono procedure di verifica che possono essere attuate dall'utente senza formattare il disco intero
- questa soluzione va bene per computer di uso "domestico"

Gestione automatica

- sono strategie più raffinate, attuate ad es. dai **controllori SCSI**
- in questo caso è il controller stesso a mantenere una lista dei blocchi difettosi, inizializzata con la formattazione fisica del dispositivo
- questa lista viene aggiornata ogni volta che il SO richiede l'accesso a un blocco che risulta difettoso, sfruttando il codice per la correzione degli errori memorizzato con i settori durante la formattazione fisica
- è possibile attuare una strategia di recupero ...

Gestione automatica

- sono strategie più raffinate, attuate ad es. dai **controllori SCSI**
- in questo caso è il controller stesso a mantenere una lista dei blocchi difettosi, inizializzata con la **formattazione fisica** del dispositivo
- questa lista viene aggiornata ogni volta che il SO richiede l'accesso a un blocco che risulta difettoso, sfruttando il codice per la correzione degli errori memorizzato con i settori durante la formattazione fisica
- è possibile attuare una **strategia di recupero** ...
- il controller tiene da parte un piccolo insieme di **settori di riserva**, che non vengono trattati come settori liberi e non vengono allocati assecondando le richieste dei processi
- per **limitare i tempi di seek**, vengono tenuti liberi alcuni settori di ciascun cilindro a cui si aggiungono un cilindro intero (o più)

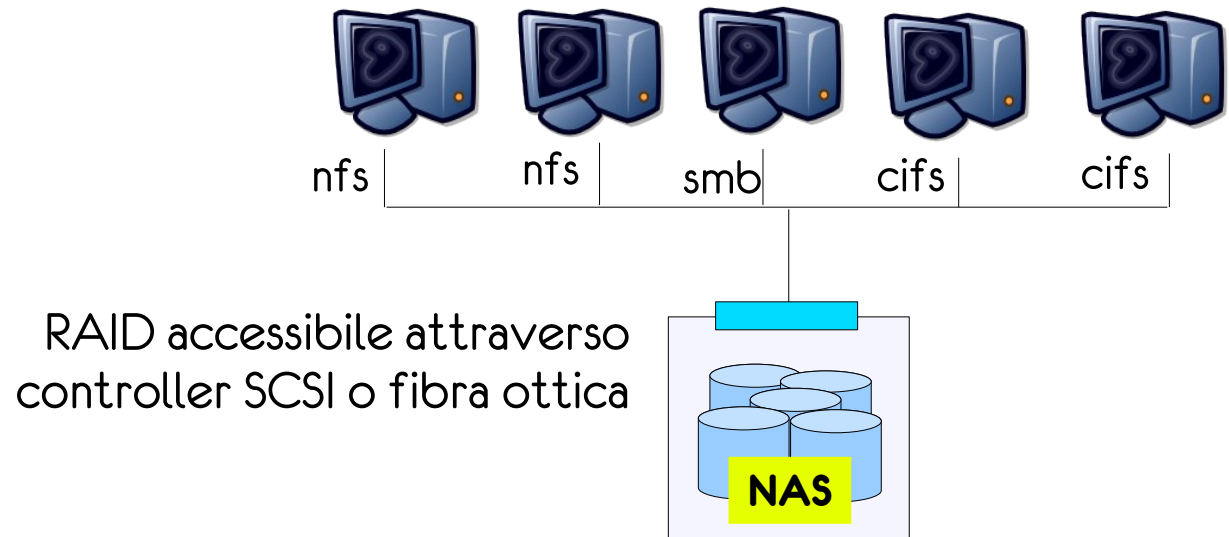


Gestione automatica

- i settori che risultano difettosi a run-time vengono **rimappati su settori accantonati**
- si cerca di utilizzare sempre **settori di riserva appartenenti allo stesso cilindro** per non stravolgere eventuali ordinamenti della coda delle richieste d'accesso
- problema: un settore difettoso conteneva probabilmente dei dati che sono andati persi, **a che serve rimapparlo su un settore vuoto?**
- ... *lo scopriremo fra poche slide ...*

RAID

- **RAID** significa Redundant Array of Independent Disks
- si tratta di una tecnologia che consente di organizzare una batteria di dischi usandola come un'unica memoria
- adatta soprattutto a realizzare file system accessibili via rete (NAS, network attached storage)



Vantaggi

- **Costo:**
 - può essere economicamente vantaggioso comperare più dischi di capacità ridotta rispetto a un disco di grande capacità;
 - i dischi possono essere aggiunti un po' per volta
- **Affidabilità:**
 - se uno dei dischi del RAID si rompe, gli altri continuano a funzionare senza problemi. Se al contrario un disco di grande capacità si rompe nessuna parte dei suoi contenuti è più accessibile
- **Ridondanza:**
 - con più dischi diventa possibile attuare tecniche di memorizzazione dei file che consentono il **ripristino automatico** dell'informazione in caso di rottura di un disco

Ridondanza

- **idea di base:** mantenere informazioni extra, non necessarie finché il sistema funziona correttamente, che però si possono sfruttare per i ripristini
- **esempio, mirroring** (o shadowing)
 - si trattano due dischi fisici come un solo disco logico
 - ogni disco fisico è l'esatta copia dell'altro
 - ogni scrittura viene effettuata su entrambi i dischi
 - per perdere i dati occorre che si rompano entrambi i dischi

