

Hardware

Integrazione all'hardware

Author: Andrea Manni

Copyright: GFDL

Version: 1.1

Indice degli argomenti

Integrazione all'hardware

Partizioni

Gestione dell'MBR

Nomenclatura dei device

I filesystem

Case

Alimentatori

Schede madri

http://it.wikipedia.org/wiki/Scheda_madre

Processori:

Architetture

IBM Cell/PPC

Embedded / NAS

RAM

Velocita' accesso dati supporti di storage

Bus per HD

RAID

RAID1: Mirror

RAID0: Stripe

RAID4:

RAID5

Mnitors

Generato con: <http://docutils.sourceforge.net/rst.html>

Nota: si consiglia la lettura di: <http://medialab.freaknet.org/sag/node29.html>

Partizioni

Ci sono 3 tipi di partizioni:

- 4 partizioni primarie
 - 1 puo' diventare estesa
 - la partizione estesa puo' contenere molte partizioni logiche

Le partizioni vengono *sicuramente* lette al momento del *Boot* dal BIOS della scheda madre (all'accensione del sistema, quando si aziona il tasto `power` per accendere il computer), quindi potrebbe essere necessario riavviare la macchina per poter vedere cambiamenti alla tabella delle partizioni.

La tabella delle partizioni risiede nei primi ~400b del disco rigido, quindi nel primo settore che chiamiamo *Master Boot Record* (MBR, primi 512 byte del supporto) nel quale e' generalmente contenuto anche un boot loader o un boot manager nel caso di sistemi operativi multipli ospitati sullo stesso supporto di storage.

Bytes	Contenuto
446 bytes	codice boot
64 bytes	tabella partizioni
2 bytes	"magic number 55AA"

Per conoscere le partizioni attualmente visibili dal Kernel in esecuzione:

```
$ cat /proc/partitions
```

Gestione dell'MBR

I dati contenuti nell'MBR sono quindi di fondamentale importanza per il sistema, dato che se questi non sono correttamente disponibili diventa impossibile accedere ai block device (e quindi ai file system e ai dati) ed eventualmente far partire il sistema (boot loader o boot manager).

E' quindi importante tenere una copia di questi dati in caso di bisogno. Il metodo piu' semplice e' l'utilizzo del comando `dd`, per copiare / riprodurre il solo schema di partizionamento (ad esempio per un sistema in RAID) si puo' usare anche `sfdisk` .

- Per modificare le partizioni si usi `# cfdisk /dev/nome_del_device`
 - `# cfdisk /dev/hda`
- Per fare una copia del contenuto dell'MBR:
 - `# dd if=/dev/hdX of=/path/MBR-backup bs=512 count=1`
- Per ripristinarlo:
 - `# dd if=/path/MBR-backup of=/dev/hdX bs=512 count=1`
- Per salvare solo la tabella delle partizioni:
 - `# dd if=/dev/hdX of=/path/backup-partition-table bs=1 count=66 skip 466`
- Per ripristinare la tabella delle partizioni:
 - `# dd if=/path/backup-partition-table of=/dev/hdX bs=1 count=66 skip 466`
- Per **cancellare completamente** tutto l'MBR:
 - `# dd if=/dev/zero of=/dev/hdX bs=512 count=1`

Nomenclatura dei device

Nei sistemi GNU/Linux la nomenclatura usata per individuare i device e le relative partizioni e' strutturata, a differenza di sistemi operativi come Windows in cui i block device vengono associati a lettere piu' o meno casuali come `C:` `D:` `E:` in base al momento in cui questi vengono resi disponibili al sistema (cosa che non permette di prevedere facilmente dove trovarli).

Tutti i device fisici sono raggiungibili nella cartella `/dev`.

Per i device di tipo EIDE / Parallel ATA su una scheda madre con due connettori si avra' a disposizione:

Primary controller:

- Primary Master: `/dev/hda`
- Primary Slave: `/dev/hdb`

Secondary controller:

- Secondary Master: `/dev/hdc`
- Secondary Slave: `/dev/hdd`

Si noti la progressione alfabetica delle lettere: `hda` e' il primo device, `hdb` il secondo e cosi' via.

Al nome del device segue il numero della partizione, es:

- `/dev/hda1` : prima partizione (Primary master EIDE)
- `/dev/hda2` : seconda partizione (Primary master EIDE)
- `/dev/hda3` : terza partizione (Primary master EIDE)
- ecc.

In uno schema *tipico* di partizionamento, nel quale si prevedono tre partizioni primarie, una estesa e alcune logiche, si avra' uno schema del tipo:

- `/dev/hda1` : prima partizione primaria (Primary master EIDE)
- `/dev/hda2` : seconda partizione primaria (Primary master EIDE)
- `/dev/hda3` : terza partizione primaria (Primary master EIDE)
- `/dev/hda5` : prima partizione logica (Primary master EIDE)
- `/dev/hda6` : seconda partizione logica (Primary master EIDE)
- `/dev/hda7` : terza partizione logica (Primary master EIDE)
- Seguono le altre eventuali partizioni logiche

Si consiglia caldamente di seguire questo schema, creando la partizione estesa nell'ultima primaria disponibile (la quarta, che per questo non compare nello schema precedente).

Si tenga presente che alcuni sistemi operativi come Windows o *BSD possono non gradire di essere collocati su partizioni logiche: si consiglia quindi di preservare le partizioni primarie per questi, eventualmente lasciando dello spazio non allocato prima della creazione delle partizioni logiche nel caso si voglia avere la possibilita' in futuro di poter installare questi sistemi. Gnu/Linux non ha nessun problema a partire da una partizione logica, mettete tranquillamente i vostri sistemi da `/dev/device5` in poi.

I filesystem

Il file system e' una struttura logica che ci permette di individuare i diversi file. Sostanzialmente e' un'insieme di meta dati che caratterizzano i singoli file piu' quanto necessario per poterli gestire.

Alcuni esempi: e' il filesystem a poter caratterizzare i file con proprietari, gruppi, permessi (lettura, scrittura, esecuzione), data di creazione e cosi' via. Alcuni filesystem permettono certe funzionalita', altri (soprattutto quelli piu' datati) sono piu' *rudimentali*.

Tipicamente i filesystem sono strutturati secondo un modello gerarchico basato su *files* e *cartelle*: quindi per poter identificare un file ci serve il suo *nome proprio* piu' il suo *percorso* nella struttura del filesystem. Il progressivo aumento del numero dei files sta cominciando a far sentire i limiti di questo modello.

Con l'aumentare delle informazioni tendono ad essere piu' efficaci modelli relazionali, che permettono di interagire coi file in modo simile a un database. Ad esempio pensiamo alle librerie che contengono centinaia di migliaia di brani musicali MP3, oppure migliaia di fotografie: cercare di gestirli tramite nome del file e cartelle diventa poco pratico. Meglio raggrupparli per autore, data, album o quant'altro si adatti alla loro natura e al loro utilizzo.

Dopo aver definito un *block device* (ad esempio avendo fatto una partizione) per poterlo usare occorre creare una struttura logica su di esso: il **file system**.

Per creare un file system di tipo ext3 si puo' usare un comando del tipo:

```
# mkfs.ext3 /dev/nome_del_device
```

Attenzione: creare un nuovo filesystem su un device che gia' ne contiene uno rende non raggiungibili i file precedentemente esistenti. Sostanzialmente potete vedere la creazione del FS come *la formattazione alla windows*.

Partizioni, nomenclatura linux, ridimensionamento, frammentazione.

Case

Antec: http://www.antec.com/ec/it/pro_enclosures.php

- tower
 - middle tower
 - full tower
- desktop
- rack per i server: <http://www.antec.com/productImage.php?ProdID=03207&SetZone=en>
- mini: es http://gizmodo.com/gadgets/images/mini_itx.jpg

Alimentatori

Voltaggi: da 60W -> ~1000. Tipicamente 400W. Si sceglie il voltaggio in base ai consumi, in particolare al giorno d'oggi le schede video (che possono consumare ~200w l'una).

Caratteristiche:

- ha una ventola di raffreddamento: quindi fa rumore e la ventola si puo' rompere col tempo.
- Sovraccarichi di tensione possono causare la rottura dell'alimentatore e qualunque altra cosa ad esso connessa. Utile usare un UPS / stabilizzatore di corrente prima dell'alimentatore. Questo vale anche per la scheda di rete / modem / RJ11.

link a esemplari in vendita: <http://www.eprice.it/default.aspx?cat3=497&des=Alimentatori>

Un buon alimentatore da PC costa circa ~80 per 400W. Non si deve risparmiare su gli alimentatori.

Per un server se ne deve sempre avere uno di scorta disponibile, o alimentazione ridondante.

Link: http://it.wikipedia.org/wiki/Alimentatore#Alimentatori_Ridondanti

Schede madri

http://it.wikipedia.org/wiki/Scheda_madre

ATX sono sempre in presa

Processori:

<http://it.wikipedia.org/wiki/CPU>

Multi core

Hyperthreading

<http://it.wikipedia.org/wiki/Hyper-Threading>

Architetture

Ci sono tante architetture e **non** sono compatibili tra di loro. Il codice oggetto e' diverso per CPU diverse.

Le vostre macchine sono generalmente della famiglia x86: tipicamente dei i686. La maggior parte del codice viene comunque compilata per i386, ma sui sistemi più recenti si può comunque utilizzare le librerie C ottimizzate per i686.

Controllate se nel vostro sistema sono presenti:

```
dpkg -l | grep libc6-i686
```

Se non sono presenti e siete su un 686 installatele!

Sulle vostre macchine gira quindi software:

- i386 : normali pc
- amd64 : i nuovi processori a 64 bit

<http://it.wikipedia.org/wiki/Microprocessore>

Link: <http://en.wikipedia.org/wiki/X86>

Elenco dei processori AMD:

- http://it.wikipedia.org/wiki/Microprocessori_AMD

Elenco dei processori Intel:

- http://it.wikipedia.org/wiki/Microprocessori_Intel
- <http://processorfinder.intel.com/details.aspx?sSpec=SLAPB>

Elenco dei processori Intel:

- <http://www.via.com.tw/en/products/processors/c7-m/>

Elenco dei processori ARM:

- http://it.wikipedia.org/wiki/Architettura_ARM - Es: Router, NAT, network appliance (fortigate...)
- <http://www.cyrius.com/debian/nslu2/>

Arm e simili (geode, Xscale, alcuni PPC) sono utilizzati per i sistemi embedded.

IBM Cell/PPC

Link: [http://it.wikipedia.org/wiki/Cell_\(processore\)](http://it.wikipedia.org/wiki/Cell_(processore))

Embedded / NAS

Devices a basse prestazioni per usi dedicati e altamente specializzati.

- <http://www.cappuccinopc.com/star-466.asp>
- <http://www.kurobox.com/>
- <http://www.linutop.com/linutop2/shop/>
- <http://www.soekris.com/>
- Meno testati: <http://www.eprice.it/default.aspx?sku=1864201>
- <http://www.onehardware.it/05/06/2008/asus-eee-box-specifiche-e-prezzi/>

RAM

http://it.wikipedia.org/wiki/Immagine:RAM_n.jpg

Velocità accesso dati supporti di storage

Tenere conto di:

- velocità teorica dei bus di connessione, es ATA / PCI
- velocità potenziale dell'hd: 7200rpm = ~45MB
- caratteristiche del FS (ottimizzazione per dimensioni file)
- numero di file e cartelle, e "vecchiaia" del FS

Bus per HD

- PATA = Parallel ATA (vecchio) - <http://it.wikipedia.org/wiki/PATA>

- SATA = Serial ATA (piu' moderno) - <http://it.wikipedia.org/wiki/SATA>
- SCSI = Small Computer System Interface - <http://it.wikipedia.org/wiki/SCSI>

RAID

- <http://www.acnc.com/raid.html>
- <http://it.wikipedia.org/wiki/RAID>

Gli hard disk hanno dei limiti fisici: dati dalle loro dimensioni finite.

Pianificare le risorse di storage,

Gli hard disks hanno la tendenza a rompersi.

La velocita' degli hard disks influenza molto le prestazioni del sistema.

Spare Disk:

Uno spare e' un hd che si lascia inattivo, in caso di rottura di uno degli elementi dell'array questo viene attivato e ne prende automaticamente il posto.

RAID1: Mirror

Piu' hard disk vengono messi in mirror, per creare ridondanza.

Caratteristiche: Modello ipotetico con hd da s=100GB con velocita' v=40MB/s

1. C'e' ridondanza dei dati: se si rompe un supporto il sistema continua a lavorare.
2. **Dimensioni: si perde meta' della capacita' di storage.**

Per 2hd si ha:

$$\text{raid size} = (s+s)/2 = 100$$

3. Prestazioni:

3.1 Lettura: Non ci dovrebbero essere cambiamenti:

$$\text{raid read speed} = v * (\text{numero array} - 1) = 120\text{MB/s} \text{ (3 hd in stripe + 1 per la parita')}$$

3.2 Scrittura: si scrive molto velocemente sulla *parte in stripe*, ma ad ogni scrittura si deve calcolare la parita' e scriverla sul supporto dedicato, che diventa il collo di bottiglia dell'intero array.

- raid read speed = dipende dell'hd della parita' = 40 MB/s

RAID0: Stripe

Piu' hard disk vengono messi in *sequenza*, per aumentare prestazioni e dimensioni massime.

Caratteristiche: Modello ipotetico con hd da s=100GB con velocita' v=40MB/s

1. Non C'e' ridondanza dei dati: se si rompe un supporto perdo tutto l'array. Non va' quindi usato per dati *importanti*, ma per dati che si possono facilmente rimpiazzare (es: una cache proxy, un mirror di qualcosa, file temporanei per la compilazione...).
2. Dimensione: si sommano le capacita' di storage.

- **Per 2hd si ha:**

- raid size= s+s = 200

- nota: s sta per la dimensione del membro piu' piccolo dell'ARRAY.

- disk1 300GB + disk2 150GB = RAID0 300GB

3. Prestazioni:

3.1 Lettura: Dovrebbero aumentare

- raid read speed = v * numero membri array = 80MB/s

3.2 Scrittura: teoricamente bisogna scrivere il doppio dei dati: le prestazioni in scrittura dimezzano.

- raid read speed = v * numer dei membri dell'array= 80 MB/s

Note:

Si usa per aumentare la velocita' e per ottenere blok device di grosse dimensioni, ma e' *pericoloso* per quanto riguarda la disponibilita' dei dati.

RAID4:

http://www.acnc.com/04_01_05.html

Piu' hard disk vengono messi in stripe con un elemento dedicato alla parita'.

Parita:

disk0	disk1	disk2	disk3
stripe	stripe	stripe	parita'
0	0	1	1
1	0	1	0
1	1	1	1
0	0	0	0
1	1	0	0

Se si rompe un HD: si puo' riapplicare l'algoritmo usato per calcolare la parita' per ri-ottenere i valori persi.

disk0	disk1	disk2	disk3
stripe	stripe	stripe	parita'
0	x	1	1
1	x	1	0
1	x	1	1
0	x	0	0
1	x	0	0

Nota: e' fondamentale avere uno spare disk disponibile: perche' in caso di rottura di due elementi perdo l'intero array senza avere possibilita' di recupero. I raid 4 o 5 tendono ad essere molto grandi, quindi il danno interesserebbe molti dati.

Caratteristiche: Modello ipotetico con hd da s=100GB con velocita' v=40MB/s

1. C'e' ridondanza dei dati: se si rompe un supporto il sistema continua a lavorare. E' fondamentale; avere uno spare disk sempre disponibile, meglio se piu' di uno.
2. Dimensioni: si perde la capacita' di storage usata per ospitare la parita'.
 - Per 2hd si ha:
 - $\text{raid size} = \text{somma (numero membri array - 1)} = 450$
3. Prestazioni: tipicamente aumentano, ma varia in base alla modalita' di gestione del RAID: se sia hardware o software.

3.1 Lettura: Dovrebbero aumentare drasticamente come per uno stripe

$\text{raid read speed} = v = 40\text{MB/s}$

3.2 Scrittura: teoricamente bisogna scrivere il doppio dei dati: le prestazioni in scrittura dimezzano.

- $\text{raid read speed} = v/2 = 20\text{ MB/s}$

Note:

La velocita' di lettura puo' aumentare: migliora notevolmente il *seek time* dato che il kernel o il controller raid puo' scegliere di leggere i dati dal'HD in cui la testina e' piu' vicina ai dati.

Inserendo uno *spare disk* si puo' aumentare ulteriormente la capacita' dell'array di sopravvivere a un guasto fisico.

RAID5

http://www.acnc.com/04_01_05.html

Il RAID5 e' un RAID4 in cui la parita' e' distribuita su tutti i membri dell'array.

Mnitors

[http://it.wikipedia.org/wiki/Monitor_\(video\)](http://it.wikipedia.org/wiki/Monitor_(video))

<http://it.wikipedia.org/wiki/HDCP>

TODO ridimensionamento, frammentazione, raid LVM