Software RAID

Termen de predare: Duminică 10 Mai, ora 23:59 (2009-05-10)

Cuprins

Enun

Să se scrie un modul de kernel care să implementeze un "software RAID". Software RAID ofera o abstractie intre dispozitivul logic si dispozitivele fizice. Implementarea va utiliza schema RAID1.

Masina virtuala dispune de doua partitii suplimentare care vor reprezenta dispozitivele fizice si un dispozitiv logic care va interfata accesul din user-space. A doua partitie va fi folosita ca backup pentru prima partitie, pentru recuperarea din eroare. Orice cerere de scriere in dispozitivul logic va rezulta in doua scrieri, cate una pentru fiecare partitie.

Fiecare partitie va stoca un sector impreuna cu o suma de control asociata (CRC32) pentru a asigura recuperarea din eroare. Daca in cazul unei citiri, un sector al primei partitii detine date corupte, se va citi sectorul de pe cea de-a doua partitie; in acelasi timp se va corecta sectorul primei partitii. In cazul in care ambele CRC-uri sunt gresite, se va returna un cod de eroare corespunzator.

Precizări generale

Pentru asigurarea recuperarii din eroare se asociază fiecărui sector un cod CRC. Codurile CRC apar după `LOG_DSK_SIZE` octei ai partiiei (definii în fiierele header pentru Linux i Windows). Structura pe disc va avea următoarea formă:

```
+-----------+-----------+-----------+      +---+---+---+
|  sector1  |  sector2  |  sector3  | .....|C1 |C2 |C3 |
+-----------+----+------------+----+--      +---+---+---+
```

`C1`, `C2`, `C3` sunt valorile CRC pentru sectoarele `sector1`, `sector2`, `sector3`. Zona CRC-urilor se regăsete imediat după `LOG_DSK_SIZE` octei ai partiiei.

Precizari Linux

- dispozitivul logic de acces va fi accesat ca un dispozitiv de tip bloc cu majorul **240** si minorul **0** sub numele `/dev/ssr`;
- cele doua discuri sunt reprezentate de dispozitivele `/dev/hdb`, respectiv `/dev/hdd`, definite prin intermediul macro-urilor `PHYS_DSK1_NAME`, respectiv `PHYS_DSK2_NAME`;
- pentru lucrul structura struct block device asociata unui dispozitiv fizic, puteti utiliza functiile open_bdev_exclusive si close_bdev_exclusive;
- pentru tratarea cererilor din user-space, se recomanda sa nu va folositi de o coada de cereri, ci sa faceti prelucrare la nivel de bio; puteti inregistra o rutina corespunzatoare folosind apelul blk_queue_make_request;
- pentru a transmite o cerere catre un dispozitiv de tip bloc, puteti utiliza functia submit_bio;
- o singură funcie de prelucrare a cererilor pentru dispozitive de tip bloc poate fi activă la un moment dat în cadrul unei stive de apeluri (mai multe detalii aici); va trebui să submitei cererile pentru dispozitivele fizice dintr-un kernel thread; se recomandă folosirea folosirea workqueues;
- pentru a transmite informatiile dorite in user-space va trebui sa asteptati terminarea lucrului cu structura struct bio transmisa dispozitivului fizic; utilizati campul bi_end_io al structurii;
- functia apelata in momentul incheierii unui bio (bi_end_io) ruleaza in context intrerupere (bottom half);
- pentru sincronizarea informatiilor scrise/citite in cadrul discului virtual cu discurile fizice (flushing), modulul va trebui sa exporte o operatie tip ioctl (`SSR_IOCTL_SYNC`); acest lucru il realizati cu ajutorul operatiilor sync_blockdev si invalidate_bdev pe dispozitivul virtual si pe dispozitivele fizice;
- pentru calculul CRC32 puteti utiliza macro-ul crc32 pus la dispozitie de kernel;
- o buna sursa de documentare o reprezinta implementarea de software RAID din kernel-ul Linux;
- cand generati structuri bio, luati in considerare ca aceasta trebuie sa aiba dimensiunea multiplu al sectorului de disc (`KERNEL_SECTOR_SIZE`);
- întrucât sectoarele de date sunt separate de sectoarele de CRC va trebui să construii structuri struct bio separate pentru date i pentru valorile CRC;
- maina virtuală dispune de două discuri suplimentare de 100 MB folosite pentru testare (`/dev/hdb` i `/dev/hdd`);
- dispozitivul virtual (`/dev/ssr`) va avea capacitatea de `LOG_DSK_SECTORS` (campul `capacity` al structurii struct gendisk);
- macrodefinitiile utile se gasesc in ssr_lin.h;
- **se recomandă** != **este obligatoriu**; orice rezolvare care respectă cerinele temei este acceptată.

Precizări Windows

- dispozitivul va fi accesat ca `\\.\SoftwareRAID`; va trebui să creai o legatură simbolică folosind IoCreateSymbolicLink;
- pentru crearea dispozitivului logic folosii apelul IoCreateDevice; tipul dispozitivului trebuie să fie `FILE_DEVICE_DISK`;
- denumirile din kernel-mode ale celor două dispozitive fizice sunt date de macrourile `PHYS_DSK_DEV1_NAME`, respectiv `PHYS_DSK_DEV2_NAME` definite în ssr_win.h;
- cele două dispozitive fizice vor fi accesate din user-mode ca `\\.\PhysicalDrive1`, respectiv `\\.\PhysicalDrive2`;
- pentru a deschide dispozitivele fizice folosii IoGetDeviceObjectPointer; nu uitai să eliberai, după utilizare, referina la dispozitiv folosind ObDereferenceObject;
- pentru a transmite o cerere read/write către dispozitivele fizice putei folosi IoBuildSynchronousFsdRequest i IoCallDriver;
- pentru calculul `CRC32` nu există `API`, dar putei utiliza implementarea crc32 de aici fără mari modificari

- în cazul în care ambele valori CRC sunt incorecte, se va returna `STATUS_DEVICE_DATA_ERROR`
- o bună sursă de documentare o reprezintă implementarea de RAM disk i exemplele de procesare a IRP-urilor;
- pentru testare se vor folosi cele două discuri IDE de dimensiune `100 MB` ataate mainii virtuale;
- pentru testare proprie trebuie să avei in vedere ca cererile trebuie să fie aliniate la 512 octei (atat offset-ul, cât i dimenisumea datelor de citit/scris);
- nu este nevoie de operaii de tip flush.

Testare

Pentru simplificarea procesului de corectare a temelor, dar i pentru a reduce greelile temelor trimise, corectarea temelor se va face automat cu ajutorul unor teste publice (linux, windows).

Testele presupun ca numele modulului de kernel este `ssr.ko` pentru Linux, respectiv `ssr.sys` pentru Windows.

Depunctări

În corectarea temei se va ine cont de următoarele criterii de depunctare:

- -1.0 warning-uri la compilare
- -1.0 implementare incorectă a cerinelor temei
- -0.2 README necorespunzător
- -0.2 memory leaks
- -0.2 omiterea eliberării resurselor (discuri, dispozitive etc.)
- -0.1 lipsă comentarii/comentarii puine/comentarii nerelevante/prea multe comentarii
- -0.1 cod neîngrijit/ilizibil/funcii kilometrice
- -0.1 pentru fiecare test picat

Întrebări

Pentru întrebări legate de temă putei consulta arhivelele listei de discuii sau putei trimite un e-mail (trebuie să fii înregistrai).