Tema 3 - Software RAID

Termen de predare: **joi, 21 Aprilie 2011, ora 23:00**

Scopul temei

Implementarea unui modul de Software RAID ce foloseşte un dipozitiv logic de tip bloc ce va citi şi scrie date de pe două dispozitive fizice, asigurând consistenţa şi sincronizarea datelor de pe cele două dispozitive fizice. Tipul de RAID implementat va fi asemănător unui RAID1.

Obiective didactice

- Înţelegerea aprofundată a modului de funcţionare a subsistemului de I/O.
- Obţinerea de deprinderi avansate de lucru cu structuri bio şi IRP.
- Acomodarea cu modul de lucru cu dispozitivele de tip bloc/disc în nucleu.
- Dobândirea de abilităţi de parcurgere şi înţelegere a codului şi API-ului dedicat subsistemului de I/O în Linux.
- Acomodarea cu API-ul de lucru cu dispozitive şi IRP-uri în Windows.

Enunţ

Să se scrie un modul de kernel care să implementeze funcţionalitatea de RAID software. Software RAID oferă o abstractizare între dispozitivul logic şi dispozitivele fizice. Implementarea va utiliza schema RAID1.

Maşina virtuală dispune de două hard disk-uri suplimentare ("Hard Disk 2(IDE)" şi "Hard Disk 3(IDE)") care vor reprezenta dispozitivele fizice. Sistemul de operare va oferi un dispozitiv (de tip bloc) logic care va interfaţa accesul din user space. Al doilea hard disk ("Hard Disk 3(IDE)") va fi folosit ca backup pentru primul "Hard Disk 2(IDE)", pentru recuperarea din eroare. Cererile de scriere în dispozitivul logic vor rezulta în două scrieri, câte una pentru fiecare hard disk. Hard disk-urile **nu** sunt partiţionate. Se va considera că fiecare hard disk are o singură partiţie ce acoperă întregul disk.

Fiecare partiţie va stoca un sector împreună cu o sumă de control asociată (CRC32) pentru a asigura recuperarea din eroare. Dacă în cazul unei citiri un sector al primei partitii deţine date corupte (valoarea CRC este greşit), se va citi sectorul de pe cea de-a doua partiţie; în acelasi timp se va corecta sectorul primei partiţii. Similar în cazul unei citiri a unui sector corupt de pe a doua partiţie. În cazul în care ambele sectoare au valori CRC greşite, se va returna un cod de eroare corespunzător.

Precizări generale

Pentru asigurarea recuperării din eroare se asociază fiecărui sector un cod CRC. Codurile CRC sunt stocate după `LOGICAL_DISK_SIZE` octeţi ai partiţiei (macro definit în fişierele header pentru Linux şi 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 regăseşte imediat după `LOGICAL_DISK_SIZE` octeţi ai partiţiei.

Precizări Linux

- dispozitivul logic va fi accesat ca un dispozitiv de tip bloc cu majorul `SSR_MAJOR` şi minorul `SSR_FIRST_MINOR` sub numele `/dev/ssr` (prin macro-ul `LOGICAL_DISK_NAME`); macro-urile sunt definite în ssr.h;
- cele două discuri sunt reprezentate de dispozitivele `/dev/sdb`, respectiv `/dev/sdc`, definite prin intermediul macro-urilor `PHYSICAL_DISK1_NAME`, respectiv `PHYSICAL_DISK2_NAME`, definite în ssr.h;
- pentru lucrul cu structura struct block device asociată unui dispozitiv fizic, puteţi utiliza funcţiile open_bdev_exclusive şi close_bdev_exclusive;
- pentru tratarea cererilor din user-space, se recomandă să nu vă folosiţi de o coadă de cereri, ci să faceţi prelucrare la nivel de bio; puteţi înregistra o rutină corespunzătoare folosind apelul blk_queue_make_request;

- pentru a transmite o cerere către un dispozitiv de tip bloc, puteţi utiliza funcţia submit_bio;
- o singură funcţie 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 submiteţi cererile pentru dispozitivele fizice dintr-un kernel thread; se recomandă folosirea workqueues;
- pentru a transmite informaţiile dorite în user-space va trebui să aşteptaţi terminarea lucrului cu structura struct bio transmisă dispozitivului fizic; utilizaţi câmpul bi_end_io al structurii;
- funcţia apelata în momentul încheierii unui bio (bi_end_io) rulează în context întrerupere (bottom half);
- pentru sincronizarea informaţiilor scrise/citite în cadrul discului virtual cu discurile fizice (flushing), modulul va trebui să exporte o operaţie tip ioctl (`SSR_IOCTL_SYNC`); acest lucru îl realizaţi cu ajutorul operaţiilor sync_blockdev şi invalidate_bdev pe dispozitivul virtual şi pe dispozitivele fizice;
- pentru calculul CRC32 puteţi utiliza macro-ul crc32 pus la dispoziţie de kernel;
- o bună sursă de documentare o reprezintă implementarea de software RAID din nucleul Linux;
- când generaţi structuri bio, luaţi în considerare că aceasta trebuie să aibă dimensiunea multiplu al sectorului de disc (`KERNEL_SECTOR_SIZE`);
- întrucât sectoarele de date sunt separate de sectoarele de CRC va trebui să construiţi structuri struct bio separate pentru date şi pentru valorile CRC;
- maşina virtuală dispune de două discuri suplimentare de 100 MB folosite pentru testare (`/dev/sdb` şi `/dev/sdc`);
- dispozitivul virtual (`LOGICAL_DISK_NAME` - `/dev/ssr`) va avea capacitatea de `LOGICAL_DISK_SECTORS` (câmpul `capacity` al structurii struct gendisk);
- macrodefiniţiile utile se găsesc în ssr.h;
- **se recomandă** != **este obligatoriu**; orice rezolvare care respectă cerinţele temei este acceptată.

Precizări Windows

- dispozitivul va fi accesat din user space ca `LOGICAL_DISK_USER_NAME` (definit în ssr.h); va trebui să creaţi un dispozitiv şi o legatură simbolică;
- pentru crearea dispozitivului logic folosiţi apelul IoCreateDevice; tipul dispozitivului trebuie să fie `FILE_DEVICE_DISK`;
- pentru crearea legăturii simbolice folosiţi IoCreateSymbolicLink);
- denumirile din kernel mode ale celor două dispozitive fizice sunt date de macro-urile `PHYSICAL_DISK1_DEVICE_NAME`, respectiv `PHYSICAL_DISK2_DEVICE_NAME` definite în ssr.h; link-urile simbolice sunt date de macro-urile `PHYSICAL_DISK1_LINK_NAME`, respectiv `PHYSICAL_DISK2_LINK_NAME`;
- cele două dispozitive fizice vor fi accesate din user mode ca `PHYSICAL_DISK1_USER_NAME`, respectiv `PHYSICAL_DISK2_USER_NAME`;
- pentru a deschide dispozitivele fizice folosiţi IoGetDeviceObjectPointer; nu uitaţi să eliberaţi, după utilizare, referinţa la dispozitiv folosind ObDereferenceObject;
- pentru a transmite o cerere read/write către dispozitivele fizice puteţi folosi IoBuildSynchronousFsdRequest şi IoCallDriver;
- pentru calculul `CRC32` nu există `API`, dar puteţi utiliza implementarea crc32 de aici fără multe modificări;
- 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 şi exemplele de procesare a IRP-urilor;
- pentru testare se vor folosi cele două discuri IDE de dimensiune `100 MB` ataşate maşinii virtuale;
- pentru testare proprie trebuie să aveţi in vedere ca cererile trebuie să fie aliniate la 512 octeţi (atât offset-ul, cât şi dimensiunea datelor de citit/scris);
- nu este nevoie de operaţii de tip flush.

Testare

Pentru simplificarea procesului de corectare a temelor, dar şi pentru a reduce greşelile temelor trimise, corectarea temelor se va face automat cu ajutorul unor teste publice (Teste Linux, Teste Windows).

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

Dacă, în urma procesului de testăre, sectoarele de pe ambele discuri conţin date nevalide, rezultând în erori de citire ce fac imposibilă funcţionarea modulului, va trebui să refaceţi cele două discuri, folosind comenzile:

```
dd if=/dev/zero of=/dev/sdb bs=8M
dd if=/dev/zero of=/dev/sdc bs=8M
```

Depunctări

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

- -1.0 implementare incorectă a cerinţelor temei
- -0.2 omiterea eliberării resurselor (discuri, dispozitive etc.)
- -0.1 pentru fiecare test important picat (`check corrupt`, `crc check`, `simple compare`)
- -0.02 pentru alte fiecare alt test picate

De asemenea, consultaţi sfaturile şi depunctările generale.

Întrebări

Pentru întrebări legate de temă puteţi consulta arhivelele listei de discuţii sau puteţi trimite un e-mail (trebuie să fiţi abonaţi).