

Operating Systems opdracht 5: ISAM

Sander van Veen & Taddeüs Kroes
6167969 & 6054129
sandervv@gmail.com & taddeuskroes@hotmail.com

12 december 2010

Inhoudsopgave

1	Inleiding tot ISAM	2
2	Implementatie van perror	3
2.1	sys_errlist[] vs. strerror()	3
2.2	strerror()	3
3	Code optimalisaties	4
4	Resultaten	4
4.1	Resultaten zonder optimalisaties	4
5	Opmerkingen	6

1 Inleiding tot ISAM

Deze opdracht van Operating Systems gaat over het efficiënt ophalen van gegevens door middel van Indexed Sequential Access Method (afgekort tot ISAM). ISAM heeft de volgende eigenschappen:

1. Data wordt opgeslagen in *fixed-length* datablokken (records), die gekenmerkt zijn door een unieke sleutel (is ook *fixed-length*).
2. Een index wordt gebruikt om de sleutels in non-lineaire tijd te doorzoeken. Dit maakt zoeken naar een specifiek datablok aanzienlijk sneller, doordat *binary search* wordt toegepast.
3. De datablokken worden sequentieel opgeslagen, gesorteerd op de waarde van de corresponderende sleutel. Dit maakt sequentieel zoeken aanzienlijk sneller.

In een ISAM-bestand worden een vast (vooraf gedefinieerd) aantal datablokken gegroepeerd opgeslagen. De datablokken in een groep worden gesorteerd op hun sleutel. De groepen krijgen elk de sleutel van het eerste datablok van de groep. Een ISAM-bestand ziet er schematisch zo uit:

```
File header - describes contents and field lengths
-----
Index records: ceil(Nblocks / 4) + ceil(Nblocks / 16) + ....
-----
Sequential data block area (Nblocks blocks)
-----
General overflow area
---- EOF -----
```

Als een datablok wordt verwijderd, wordt het datablok niet uit het ISAM-bestand verwijderd. In plaats van het verwijderen uit het ISAM-bestand, wordt het datablok enkel als *deleted* gemarkeerd. Op het moment dat er een nieuw datablok op de plek van een eerder verwijderd datablok wordt geplaatst, wordt het datablok naar de hardeschijs geschreven. Logischerwijs wordt de *deleted* markering van het weggeschreven blok dan weggehaald. De index houdt dan bij welke blokken als *deleted* zijn gemarkeerd. Dit reduceert het aantal disk writes aanzienlijk. Het gevolg is dat veel *random delete* en *random insert* acties sneller kunnen worden uitgevoerd. Want de hardeschijs vormt het grootste bottleneck van een database.

Doordat de verwijderde datablokken niet uit het ISAM-bestand worden verwijderd, kan het voorkomen dat een grote hoeveelheid nutteloze data in het ISAM-bestand achterblijft. Het opschonen van het ISAM-bestand zou daarom periodiek moeten worden uitgevoerd. Het gunstigste is om dit gedurende de daluren te doen.

De *General overflow area* van een ISAM-bestand maakt het mogelijk om (theoretisch) oneindig veel datablokken op te slaan. In de praktijk is dit niet haalbaar doordat er niet een oneindige hoeveelheid schrijfruimte bestaat. Bij *compile-time* kan niet worden gezien wat de lengte van de sleutels wordt. Daarom wordt er gebruik gemaakt van *array-indexing* en *casting*.

Tot slot is er nog op te merken dat het toevoegen van een datablok met een kleinere sleutel dan al de bestaande datablokken, wordt gedaan door de *empty string* sleutel te gebruiken. De datablokken worden immers in groepen ingedeeld en zonder deze *empty string* sleutel zou het niet mogelijk zijn om een datablok met een lagere sleutel op te slaan aan het begin van de eerste groep.

2 Implementatie van perror

De functie `perror()` wordt geïmplementeerd door een lijst met foutbeschrijvingen te doorzoeken. De globale variabele `errno` wordt gebruikt om de gewenste foutmelding te vinden. Het gebruik van deze lijst (genaamd `sys_errlist[]`) wordt afgeraden omdat deze lijst niet op alle platformen beschikbaar is. Als alternatief wordt `strerror()` gegeven.

2.1 `sys_errlist[]` vs. `strerror()`

Om een programma te kunnen compileren op diverse platformen, wordt het aangeraden om `strerror()` te gebruiken. Dit volgt onder andere uit het onderstaande citaat van de GCC maintainers:¹

If a program has only support for `sys_errlist[]` you will have to do some work to make it compile on GNU, which has dropped support for it and does only provide `strerror()`. Steinar Hamre writes that `strerror()` should be used because:

1. It is the modern, POSIX way.
2. It is localized.
3. It handles invalid signals/numbers out of range. (better errorhandling and not a buffer-overflow-candidate/security risk)

`strerror()` should always be used if it is available. Unfortunately there are still some old non-POSIX systems that do not have `strerror()`, only `sys_errlist[]`.

Today, only supporting `strerror()` is far better than only supporting `sys_errlist[]`.

2.2 `strerror()`

Naast de *portability* eigenschap van `strerror()`, maakt de function het ook mogelijk om de voorgedefinieerde foutomschrijvingen van de C library te gebruiken. Uit de opdracht kon niet worden opgemaakt of deze functie mocht worden gebruikt, dus is er voor gekozen om een nieuwe functie genaamd `isam_strerror` aan te maken. Door middel van `_user_strerror` kunnen er meer foutbeschrijvingen worden toegevoegd aan `strerror()`.² De functie `_user_strerror` is verder niet gebruikt, doordat het aangeleverde framework niet de function `perror()` aanroept.

¹Use `strerror()`: <http://www.gnu.org/software/hurd/hurd/porting/guidelines.html>

²Gebruik van `_user_strerror`: <http://www.slac.stanford.edu/comp/unix/package/rtems/doc/html/libc/libc.info.strerror.html>

3 Code optimalisaties

De oorspronkelijke functie `isam_readByKey()` was inefficiënt geprogrammeerd, doordat er onnodig werk werd gedaan. De oorspronkelijke functie roept de functie `isam_setKey()` aan gevolgd door de functie `isam_readNext()`. In plaats van direct het datablok terug te geven, wordt er eerst achteruit gezocht naar het datablok.

De tweede optimalisatie is terug te vinden in de functie `isam_update()`. Deze functie was inefficiënt doordat er twee keer naar een datablok werd gezocht: eerst om het blok te verwijderen en vervolgens nog een keer om het datablok te overschrijven. Als het datablok direct wordt overschreven (zonder dus te verwijderen), scheelt dit lezen van de hardeschijf.

Om de optimalisaties te vergelijken, is een *preprocessor directive* aangemaakt die de optimalisaties aan of uit zet. Deze directive heeft de naam `OPTIMISED`. Door in een shell deze directive aan te maken (`export OPTIMISED=1`) en het test programma opnieuw te compileren (`make clean; make`), worden de optimalisaties toegepast.

4 Resultaten

4.1 Resultaten zonder optimalisaties

Allereerst (chronologisch geordend) worden de resultaten van de implementatie zonder optimalisatie behandeld. Om de prestatie te kunnen meten, worden de volgende instructies uitgevoerd in een shell:

```
$ unset OPTIMISED
$ rm klant.isam
$ make clean && make -s
$ ./isam_test klant.isam < tele &> /dev/null
$ ./isam_bench namen initialen titels
```

De volgende resultaten worden verkregen met 6 als waarde voor de cache grootte. Dit is de standaard waarde voor de cache grootte in deze opdracht.

```
Lees namen van namen
Lees initialen van initialen
Lees titels van titels
Failed to create file: File already exists (3)
Bestaand bestand met 488 records geopend
Stats after opening isam file...
calls: 489
reads: 42
writes: 0
hwrites: 0
```

```
Er zijn 0 records gelezen; mailings zijn verzonden aan 0
500 records gelezen, 0 mislukt
200 nieuwe klanten, 171 vertrokken
Er zijn 42 records gelezen; mailings zijn verzonden aan 42
```

500 records gelezen, 125 mislukt
200 nieuwe klanten, 142 vertrokken
Er zijn 78 records gelezen; mailings zijn verzonden aan 78
500 records gelezen, 173 mislukt
200 nieuwe klanten, 113 vertrokken
Er zijn 429 records gelezen; mailings zijn verzonden aan 429
Er zijn 429 records gelezen; mailings zijn verzonden aan 429
Stats after Dick Albada's benchmark...
calls: 696340
reads: 472666
writes: 8672
hwrites: 5999

Stats after benchmarking:
1000 readByKey
400 update
100 writes
100 delete

calls: 546289
reads: 356770
writes: 1994
hwrites: 1552

Stats after entire benchmark:
calls: 1243607
reads: 829520
writes: 10666
hwrites: 7551

Daarna is de cache grootte veranderd naar 2 en is gekeken wat de resultaten zijn:

Lees namen van namen
Lees initialen van initialen
Lees titels van titels
Failed to create file: File already exists (3)
Bestaand bestand met 488 records geopend
Stats after opening isam file...
calls: 489
reads: 43
writes: 0
hwrites: 0

Er zijn 0 records gelezen; mailings zijn verzonden aan 0
500 records gelezen, 0 mislukt
200 nieuwe klanten, 171 vertrokken
Er zijn 42 records gelezen; mailings zijn verzonden aan 42
500 records gelezen, 125 mislukt
200 nieuwe klanten, 142 vertrokken

Er zijn 78 records gelezen; mailings zijn verzonden aan 78
500 records gelezen, 173 mislukt
200 nieuwe klanten, 113 vertrokken
Er zijn 429 records gelezen; mailings zijn verzonden aan 429
Er zijn 429 records gelezen; mailings zijn verzonden aan 429
Stats after Dick Albada's benchmark...
calls: 696357
reads: 559525
writes: 8672
hwrites: 5999

Stats after benchmarking:
1000 readByKey
400 update
100 writes
100 delete

calls: 546290
reads: 405557
writes: 1994
hwrites: 1552
Stats after entire benchmark:
calls: 1243625
reads: 965168
writes: 10666
hwrites: 7551

5 Opmerkingen