LUXOR Datorer

Anvisning

ISAM 800

ABC800®

1 Allmänt om ISAM

ISAM till ABC800 är hjälppaket som innehåller stöd för att skapa olika former av register. ISAM innehåller även ett antal nya BASIC-instruktioner. Med hjälp av ISAM kan ABC800 i en Multi-User miljö (fleranvändarmiljö ABC-NET) som jobbar mot samma fil.

ISAM är en metod att organisera register för att hitta enskilda poster i stora register. I ett medlemsregister kan du söka med hjälp av t.ex. ett medlemsnummer, i ett artikelregister kan du söka med hjälp av t.ex. artikelnumret eller varugruppskoden.

När du använder ISAM är dessutom posterna alltid sorterade. Därför kan du lätt skriva ut en lista sorterad i bokstavs- eller nummerordning.

ISAM betyder Index-Sekventiell Access Metod vilket innebär att man med hjälp av ett index hittar poster i en "vanlig" sekventiell fil. Detta förklaras enklast med hjälp av ett exempel.

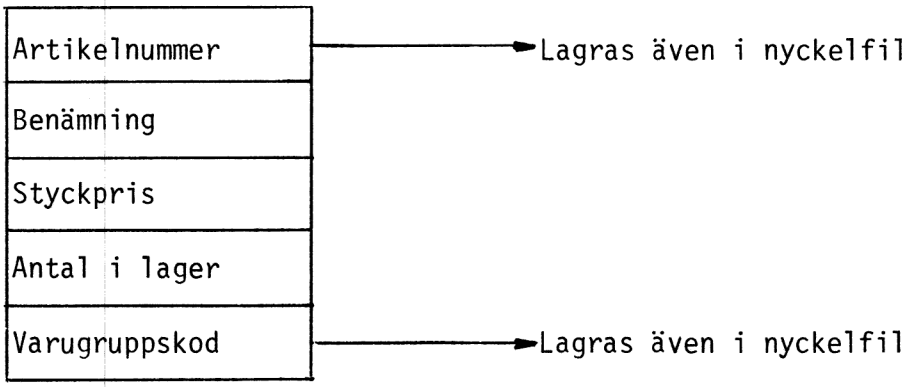
Antag att du har ett artikelregister med följande uppgifter:

1 Artikelnummer
2 Benämning
3 Styckpris
4 Antal i lager
5 Varugruppskod

Du vill kunna söka artiklarna dels med hjälp av artikelnumret och dels med hjälp av varugruppskoden.

Utan ISAM skulle du antagligen vara tvungen att leta igenom alla artiklar innan du hittar den du söker. Ju större register desto längre tid skulle det ta.

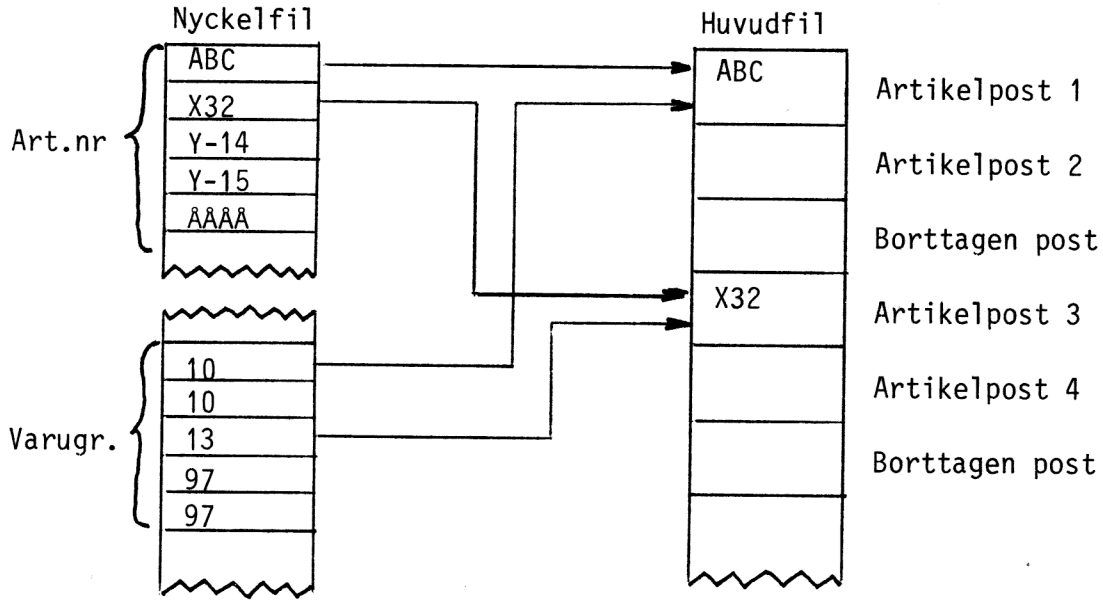
ISAM klarar detta genom att lägga upp två filer. Den första innehåller alla de fem uppgifterna om varje artikel. Den andra innehåller bara de två sökbegreppen (nycklarna) samt adresspekare till den första filen.



När du söker efter en artikel med ett visst artikelnummer söker datorn först upp numret i nyckelfilen. Där hämtar den en adress-pekare till huvudfilen. Slutligen går den till huvudfilen och läser in hela posten. För att det ska gå riktigt snabbt är nyckelfilen sorterad. Sorteringsmetoden kallas B-träd vilket garanterar att filen alltid utnyttjas på ett optimalt sätt. Du behöver aldrig göra några tidskrävande reorganiseringar av dina filer.

Du kan ha flera olika index till varje post. I det här fallet var det dels artikelnumret och dels varugruppskoden som användes som index. Maximalt kan varje post sökas med hjälp av tio olika index.

Du bör dock försöka programmera med så få index som möjligt. Dels kräver de mycket utrymme på flexskivan och dels tar det längre tid att mata in nya poster ju fler index du använder.



I det här exemplet har vi dels index ARTNR med nycklarna ABC, X32, Y-14, Y-15 samt ÅÅÅÅ och dels index VARUGR med nycklarna 10, 10, 13, 97 samt 97. Artikelpost nummer 3 kan antingen sökas med hjälp av nyckeln X32 i index ARTNR eller med nyckeln 13 i index VARUGR.

2 Laddning av ISAM-rutinerna

För att kunna använda ISAM måste du sätta programskivan med ISAM i flexskiveenheten och trycka på RESET. Därmed laddas ISAM-rutinerna in. Dessa kräver ungefär 5k bytes av primärminnet.

Nedanstående filer finns på ISAM-skiva för att ladda in ISAM-funktionerna. De måste kopieras till alla skivor som du använder för att starta systemet. Däremot behöver de inte finnas på dataskiva.

ADDOPT.ABS — Laddar ISAM och eventuella andra optioner. Med ADDOPT.ABS är det möjligt att ladda flera relokerbara filer (optioner ISAM eller/och RAM-floppy). Detta görs genom att filnamnen placeras i ADDOPT-filen på följande sätt:

```
10 OPEN "ADDOPT.ABS" AS FILE 1
20 PUT #1,CHR¤(255,255)+"FILNAMN1REL"+CHR¤(255)+
   "FILNAMN2REL"+.....+CHR¤(254)
30 CLOSE 1
```

Filnamn måste anges med stora bokstäver, åtta tecken filnamn och med tre tecken filtyp. Som regel gäller att alla relokerbara filer har filtyp REL.

OBS!
Filnamn och filtyp skall inte åtskiljas med . (punkt).

Då endast drivrutinen för ISAM skall laddas, skall ADDOPT.ABS innehålla filnamnet ISAMOPT REL. Den bifogade ADDOPT-filen innehåller detta namn.

ISAMOPT.REL — Innehåller program för ISAM.

BASICINI.SYS — Motsvarar den ordinarie BASICINI-filen men innehåller dessutom felmeddelanden för ISAM.

Flexskivan innehåller dessutom ytterligare två filer som behövs vid programmeringen. De behöver dock inte kopieras till varje programskiva när programmet är klart. De heter:

CREINDEX.BAC — Skapar en nyckel- och en huvudfil. Detta sker i dialog med programmeraren.

PREABS.BAC — Motsvarar det ordinarie PRESTART-programmet men styr dessutom att ISAM-rutinerna laddas. Dvs laddar både ABS-fil och BASCI-fil.

Slutligen kan skivan innehålla en fil som bara behövs om du vill ändra ordningen vid ASCII-sortering. Normalt sorteras siffror in före bokstäver och stora och små bokstäver likställs.

SORTORDR.TAB — Innehåller en tabell som anger hur varje tecken ska sorteras. Se bilaga 1.

3 Numerisk sortering och sortering tecken-för-tecken

All lagring sker i form av strängar. Varje post motsvarar en sträng. Denna kan i sin tur bestå av flera strängar där varje delsträng motsvarar ett fält. Om du ska lagra tal måste du först omvandla dem till strängar med hjälp av funktionen CVT%¤ för heltal eller CVTF¤ för flyttal. Därefter kan du placera in dem i huvudsträngen med hjälp av funktionen MID¤.

Alla poster måste vara lika långa och de fält som utgör index måste börja på bestämda platser inom strängen.

Nycklarna kan sorteras på olika sätt. Ett sätt är en ren tecken-för-tecken sortering från vänster till höger. Det är så som namnen är sorterade i telefonkatalogen.

Ett annat sätt är en numerisk sortering. Då sorteras alla tal i storleksordning. Skillnaden kan visas med ett exempel:

Vi vill sortera talen 11, 9 och 10.

Vid en numerisk sortering placeras de naturligtvis i ordningen:

```
 9
10
11
```

Vid en tecken-för-tecken sortering ser man bara på ett tecken i taget. Om det första tecknet är samma tittar man på det andra tecknet osv. Ordningen blir då:

```
10
11
9
```

För varje index måste du tala om vilken typ av sortering som ska göras.

4 Skapa ISAM-filer

För att kunna använda ISAM måste du skapa en nyckel- och en huvudfil. Det görs enklast med programmet CREINDEX som frågar dig hur filerna ska se ut.

Innan du startar programmet med kommandot RUN CREINDEX bör du noga tänka igenom hur du vill att dina poster ska se ut, vilka fält som ska ingå och vilka som ska vara nycklar.

```
** Skapa ISAM-filer Ver 1.0 **

     * Skapa filer *

Namn på nyckelfil ? _______
Namn på huvudfil  ? _______
Postlängd         ? ___
```

Filnamnen anges utan filtyp. Filtypen kommer automatiskt att sättas till .ISM för nyckelfilen och .DAT för huvudfilen. Du bör ange samma namn på båda filerna. När du senare ska öppna filerna behöver du bara öppna nyckelfilen - huvudfilen öppnas automatiskt.

Postlängden avser antalet tecken för varje post i huvudfilen. Varje post måste alltså ha en fast längd.

För varje index ska du därefter ange:

```
** Skapa ISAM-filer Ver 1.0 **

      * Skapa index nr 1 *

Namn på index          ? ________
Startposition          ? __
Längd av index         ? __
Indextyp (B,A,I,F,D)   ? _
Dubblett-nycklar (J/N) ? _
```

Varje index ges ett namn på högst åtta tecken. Det kan vara t.ex. ARTNR, VARUGR, POSTNR eller FÖRNAMN. Senare när du vill söka en post med ett visst index är det detta namn du ska refererа till.

Startposition och längd av index anger i vilken position i huvudsträngen som indexet startar och hur många tecken som indexet ska omfatta.

Det finns fem olika indextyper:

B — Binär. Tecken-för-tecken sortering.
Hela strängen betraktas som ett positivt binärt heltal med mest signifikanta tecken först.

A — ASCII. Tecken-för-tecken sortering.
Denna ska du använda vid sortering av allt som kan innehålla annat än tal. Normalt sorteras alla tecken i strikt bokstavsordning utan någon skillnad mellan stora och små tecken. Denna ordning kan dock ändras. Se bilaga 1.

I — Integer. Numerisk heltalssortering.
Längd av index måste sättas till 2. Heltalet placeras in i strängen med hjälp av funktionen CVT%¤(I%).

F — Flyttal. Numerisk sortering.
Längd av index måste sättas till 4 bytes. Talet placeras in i strängen med hjälp av funktionen CVTF¤(A). Programmet ska vara inställt för SINGLE precision.

D — Dubbel precisions flyttal. Numerisk sortering.
Längd av index måste sättas till 8 bytes. Talet placeras in i strängen med hjälp av funktionen CVTF¤(A). Programmet ska vara inställt för DOUBLE precision.

På frågan om dubblett-nycklar ska du svara J eller N för att tala om om programmet ska acceptera att nycklarna för två olika poster kan vara lika. I exemplet med artikelregistret svarar du N för artikelnumret eftersom detta måste vara unikt. För varugruppsnumret svarar du J eftersom flera artiklar kan tillhöra samma varugrupp.

Om du har svarat N på frågan om dubblett-nycklar och sedan försöker mata in en ny post med samma nyckel som en befintlig ges ett felmeddelande.

Dessa frågor ska besvaras för varje index. När du inte vill ange fler index trycker du bara RETURN på frågan om namn på index.

5 Exempel på hur en ISAM-fil skapas

Vi ska gå igenom detta med hjälp av exemplet från sidan 1, där vi hade ett enkelt artikelregister. Vi börjar med att göra en tabell över alla fält i posten. Det är bara de fält som ska utgöra index, som behöver fyllas i fullständigt:

Beskrivning	Namn	Typ	Längd	Startposition	Dubblett
Artikelnummer	ARTNR	A	7	1	N
Benämning		A	25	8	
Styckpris		F	4	33	
Antal i lager		F	4	37	
Varugruppskod	VARUGR	I	2	41	J
			42		

Dialogen i programmet CREINDEX kan se ut så här (dina svar är angivet efter varje frågetecken):

```
** Skapa ISAM-filer Ver 1.0 **
      * Skapa filer *

Namn på nyckelfil ? ARTIKLAR
Namn på huvudfil  ? ARTIKLAR
Postlängd         ? 42

      * Skapa index nr 1 *
Namn på index           ? ARTNR
Startposition           ? 1
Längd av index          ? 7
Indextyp (B,A,I,F,D)    ? A
Dubblett-nycklar (J/N) ? N

      * Skapa index nr 2 *
Namn på index           ? VARUGR
Startposition           ? 41
Längd av index          ? 2
Indextyp (B,A,I,F,D)    ? I
Dubblett-nycklar (J/N) ? J

      * Skapa index nr 3 *
Namn på index           ? <RETURN>
```

Därefter skapas filerna och är färdiga att användas.

6 Nya BASIC-instruktioner

När du har laddad in ISAM har du tillgång till följande extra BASIC-instruktioner:

ISAM OPEN	Öppnar en ISAM-fil för skrivning, läsning etc.
ISAM WRITE	Skriver en ny post i en ISAM-fil
ISAM READ	Läser en befintlig post i en ISAM-fil
ISAM UPDATE	Ändrar en post i en ISAM-fil
ISAM DELETE	Tar bort en post i en ISAM-fil

De olika kommandona beskrivs i tur och ordning.

6.1 ISAM OPEN

En ISAM-fil måste alltid öppnas med ISAM OPEN innan du kan skriva eller läsa i den. Om du öppnar den med bara OPEN finns det en risk att du förstör den.

```
ISAM OPEN "<enhet:>filnamn.typ" AS FILE nr
```

Syntaxen är exakt identisk med den vanliga OPEN-instruktionen.

Observera att du ska ange namnet på index-filen och inte på huvudfilen. Om du inte anger någon filtyp kommer denna automatiskt att sättas till .ISM.

Du stänger filen igen med CLOSE nr.

Exempel:

```
100 ISAM OPEN "ARTIKLAR" AS FILE 5
.
.
900 CLOSE 5
```

6.2 ISAM WRITE

Detta är det enda sätt som får användas vid insortering av ny post i ISAM-filen.

```
ISAM WRITE # nr,sträng
```

Strängen innehåller hela posten som ska lagras. Dvs längden på varje fält måste utfyllas till maximal längd som sedan läggs samman till en total sträng. ISAM rutinen vet vilken eller vilka delar av strängen som utgör index.

Skrivning sker i både huvudfilen och indexfilen.

Om en dubblett förekommer i ett index där det inte är tillåtet avbryts skrivningen med ett felmeddelande (nr 121). Varken huvud-

filen eller indexfilen kommer då att påverkas.

Exempel:

```
100 ISAM OPEN "ARTIKLAR" AS FILE 5
110 A¤="X37     Nya Produkten                          "
120 ISAM WRITE #5,A¤
130 CLOSE 5
```

6.3 ISAM READ

Denna instruktion kan användas både vid sökning efter en speciell nyckel och vid utskrifter i sorterad ordning. De parametrar som står inom klammer-parentes är frivilliga.

```
ISAM READ # nr,strängvariabel1 <INDEX sträng2> <KEY sträng3>
                                                 <FIRST>
                                                 <LAST>
                                                 <NEXT>
                                                 <PREVIOUS>
```

Resultatet av läsningen kommer att placeras i strängvariabel1.

Läsningen sker med det index som bestäms av INDEX sträng2. Här anger du samma namn som när du skapade indexfilen. Det kan t.ex. vara ARTNR, VARUGR, POSTNR eller FÖRNAMN.

Om du inte anger något index kommer det senast använda indexet att användas. Om det är första gången som läsning sker i filen kommer indexfilens första index att användas.

Sökning efter en speciell nyckel sker genom att du anger KEY sträng3. Om nyckeln finns läses hela posten in. Om nyckeln inte finns ges felmeddelande 120. Om det finns flera poster med samma nyckel kommer den första posten att läsas in. De övriga läses med NEXT.

Istället för KEY kan något av argumenten FIRST, LAST, NEXT eller PREVIOUS användas. De läser in första, sista, nästa resp. föregående post enligt nyckelns sortering. Nästa och föregående är alltid i förhållande till senaste läsning eler skrivning i filen.

Om du inte anger någon nyckel kommer det att tolkas som NEXT.

Vid läsning med KEY behöver du inte ange hela sökbegreppet. Sökordet PER kan alltså hitta både PER och PERSSON. Om du vill vara säker på att bara hitta det första måste du alltid fylla ut sökordet med blanka tecken på slutet både vid skrivning och läsning.

Exempel 1:

```
ISAM READ #1,A¤ INDEX "NAMN" KEY "PER"
```

Söker i index "NAMN" efter den första posten med nyckeln "PER" (eller den första nyckel som börjar med "PER"). Posten läses in till variabeln A¤.

Exempel 2:

ISAM READ #1,A¤ NEXT

Läser in nästa post (enligt samma index som föregående gång).

Exempel 3:

ISAM READ #1,A¤ INDEX "ARTNR"

Läser in den första posten enligt index "ARTNR".

Exempel 4:

```
100 ISAM OPEN "KUNDER" AS FILE 5
110 I¤="NAMN"
120 ISAM READ #5,A¤ INDEX I¤ FIRST
130 PRINT A¤
140 ON ERROR GOTO 180
150 ISAM READ #5,A¤ INDEX I¤ NEXT
160 PRINT A¤
170 GOTO 150
180 CLOSE 5
190 END
```

Skriver ut en lista över alla poster sorterad enligt index "NAMN". Programmet kan förkortas genom att du tar bort raderna 120 och 130 samt tar bort ordet NEXT från rad 150.

6.4 ISAM UPDATE

Ändrar en befintlig post i ett register. Alla index ändras automatiskt om de har förändrats.

```
ISAM UPDATE # nr, sträng1 TO sträng2
```

Sträng1 måste innehålla resultatet från föregående läsning i denna ISAM-fil.

Sträng2 innehåller den nya sträng, som ska skrivas i filen i stället för sträng1.

Observera att du först måste göra ISAM READ på en post innan du kan göra ISAM UPDATE på den. Om medskickad Sträng1 ej överensstämmer med posten frånföregående läsning ges felmeddelande 123. Detta används för att kontrollera uppdatering i ett Multi-User system.

Om postens nya innehåll medför att någon av nycklarna har ändrats kommer även nyckelfilen att påverkas. Om det därvid uppkommer en dubblett i ett index som inte får ha dubbletter, avbryts operationen med felmeddelande nr 121. Postens gamla innehåll kommer då att ligga kvar oförändrat i filen.

Exempel:

```
100 ISAM OPEN "ARTIKLAR" AS FILE 5
110 ISAM READ #5,A¤ INDEX "ARTNR" KEY "X32"
120 B¤=A¤ : MID¤(B¤,8,9)="Nytt namn"
130 ISAM UPDATE #5,A¤ TO B¤
140 CLOSE 5
```

Här läses en post med artikelnummer "X32". Posten modifieras och skrivs tillbaka med ISAM UPDATE.

6.5 ISAM DELETE

Med instruktionen kan en post tas bort ur ISAM-filen. Det lediga utrymmet kan sedan användas för inmatning av nya poster.

```
ISAM DELETE # nr, sträng1
```

Sträng1 måste innehålla resultatet från föregående läsning i filen.

Observera att du först måste göra ISAM READ på en post innan du kan göra ISAM UPDATE på den. Om medskickad Sträng1 ej överensstämmer med posten frånföregående läsning ges felmeddelande 123. Detta används för att kontrollera uppdatering i ett Multi-User system.

Exempel:

```
100 ISAM OPEN "KUNDER" AS FILE 5
110 I¤="NAMN" : K¤="PER"
120 ISAM READ #5,A¤ INDEX I¤ KEY K¤
130 ISAM DELETE #5,A¤
140 CLOSE 5
```

Den första posten med nyckeln "PER" i index "NAMN" läses och tas bort.

7 Felmeddelanden

Felmeddelande nr 120 - 129 är speciella för ISAM-rutinerna. Filen BASICINI på ISAM-skivan har kompletterats med nedanstående feltexter:

120 Nyckel finns ej
121 Dublett nyckel
122 Felaktig nyckel
123 Fel vid kontrolläsning
124 Index finns ej
125 Felaktig postlängd
126 Fel isam-fil version
127 Ej använd felkod
128 Slut på minne i centralen
129 Reserverad felkod

Bilaga 1 - Sorteringsordning vid ASCII-sortering

Om du anger indextyp = A för ett index kommer nycklarna att sorteras i strikt bokstavsordning. Det betyder att sorteringen kommer att följa ASCII-tabellen (Se BASIC-handboken) med ett par undantag:

1 Stora och små bokstäver likställs. I ASCII-tabellen följer alla små bokstäver efter alla stora.

2 Å, Ä och Ö sorteras i rätt ordning. I ASCII-tabellen är de omkastade!

Denna ordning kan ändras genom att du lägger till en fil med namnet SORTORDR.TAB. När ISAM-programmet laddas kommer också denna fil att läsas.

I filen anges hur de olika tecknen ska sorteras. Observera att du inte får lägga till denna fil, eller göra ändringar i den, efter det att Du har börjat mata in poster. Då kan du nämligen förstöra ISAM-filen!

SORTORDR.TAB ska innehålla 128 bytes. Varje byte motsvarar en av positionerna 0 - 127 i ASCII-tabellen. I denna position ska sedan ett tal mellan 0 och 255 skrivas. Detta tal anger den vikt som tecknet ska ges vid sorteringen.

Ett exempel:

Bokstaven A har nummer 65 i ASCII-tabellen. Normalt har den också vikten 65. Det betyder att i position 65 står det 65. Om du vill ändra vikten för bokstaven A så att den t.ex. blir likställd med bokstaven C vid sorteringen ska du skriva in talet 67 i position 65.

Bilaga 2 - Data för ISAM

Flexskive utrymme

Storleken på datafilen är beroende på antalet poster i filen och längden på posten:

Sd = Ad * Pd

där Sd är storleken på datafilen

Ad är antalet poster

Pd är postlängden

Storleken på ISAM-filen är beroende på antalet poster i filen, antalet index och längden på varje nyckelsträng. Alla poster har en nyckel för varje index. Varje nyckel innehåller en nyckelsträng och en pekare. Utrymme för alla nycklar beräknas enligt följande:

$$S = Ad * \sum_{i} (Ni + Np + 1)$$

där Ad är antalet poster

Ni är längden på nyckelsträng nr. i

Np är storleken på nyckelpekaren (3 byte)

Enligt principen för B-träd ger detta i medelvärde 75% (50% i sämsta fall) av filen kommer att utnyttjas för nycklar. Den aktuella storleken på isamfilen blir approximativt:

SI = 4/3 * S (Medelvärde)

eller

SI = 2 * S (Sämsta fall)

Hastighet

Tiden för att hämta en slumpmässig post genom att använda ISAM är beroende av:

a. Antal nivåer på index-trädet.

b. Storleken på filerna.

c. Accesstiden för flexskiveenheten.

I princip krävs det en skivaccess för varje nivå i indexträdet plus en access för att hämta posten. Antalet nivåer i indexträdet beror på antalet nycklar som finns i varje sektor och totalt antal nycklar i trädet.

$$n = \frac{Ri-FP-SP-2}{Ni+Np+SP+1}$$

Rotsektorn för varje index på trädet innehåller mellan 1 och n nycklar. Alla övriga sektorer innehåller mellan n/2 och n nycklar. Det finns en extra son pekare i varje sektor. I sämsta fall innehåller roten en nyckel och övriga sektorer n/2 nycklar. I detta fall kan trädet ses som två under träd, som vardera innehåller (Ad-1)/2 nycklar. Antalet nivåer blir:

$$h <= 1 + \frac{LN((Ad-1)/2)}{LN(n/2+1)} \qquad \text{(Sämsta fall)}$$

I bästa fall blir det:

$$h >= \frac{LN(Ad)}{LN(n+1)}$$

För att läsa en data post med hjälp av en nyckel krävs det h+1 skivaccesser.

Storleken på filen påverkar den fysiska söktiden på skivan och graden av "overhead" som krävs för att hitta skivsektorn i den logiska filen.

Söktiden kommer att vara direkt proportionell mot medeltiden för den fysiska enheten.

Förkortningar

Sd = Totalt utrymme för datafilen
Ad = Antalet dataposter
Pd = Postlängden
Ni = Längd på nyckelsträng nr. i
Np = 3 byte (Storlek på nyckelpekaren)
S = Antal tecken i ISAM-filen
SI = Utrymme som krävs på skivan, för ISAM-filen
n = Antalet nycklar som ryms i varje sektor
Ri = 253 byte (fysisk sektorformat)
FP = 3 byte (Father pointer)
SP = 3 byte (Son pointer)
h = Nivåer i indexträdet

Art. nr. 66 79206-01