

Tomáš Oberhuber

Faculty of Nuclear Sciences and Physical Engineering
Czech Technical University in Prague

VSAM – Virtual Storage Access Method

1 VSAM – Virtual Storage Access Method

2 Program IDCAMS

- jde o speciální soubory určené pro databázové aplikace – IMS, DB2
- přidávají určitou organizační strukturu k záznamům
- rozlišujeme několik typů VSAM souborů

- **KSDS – Key Sequence Data Set**
 - zřejmě nejčastěji používaná metoda
 - každý záznam má 1 nebo více klíčů, pomocí kterých lze k záznamu přistupovat
 - záznamy mají proměnnou délku
- **ESDS – Entry Sequence Data Set**
 - záznamy jsou uloženy sekvenčně, lze k nim přistupovat
 - sekvenčně
 - přímo pomocí RBA - relative byte address (typ long unsigned int)
 - pomocí indexů
- **RRDS – Relative Record Data Set**
 - k záznamům se přistupuje pomocí čísel
 - REC1, REC2,
- **LDS – Linear Data Set**
 - byte-stream - posloupnost bajtů
 - využívají je některé systemové funkce z/OS

Struktura VSAM souborů

VSAM pracují s tzv.

- CI – *Control Interval*
- CA – *Control Area*

Struktura CI



- RDF – *Record Descriptor Field*
 - ukazuje, kde začíná a končí daný záznam
- CIDF – *Control Interval Descriptor Field*
 - obsahuje informaci o volném místě

CI se skládají do CA

Struktura VSAM souborů

Příklad vložení záznamu pro KSDS

- záznamy jsou uloženy tak, že jsou seříděny vzestupně
- každý CI obsahuje všechny záznamy mezi nejmenší a největší hodnotou klíče v daném CI
- tak lze snadno určit, do kterého CI se má daný záznam vložit
- pokud je v CI dostatek volného místa, záznam se vloží tak, aby všechny záznamy v CI byly opět vzestupně seřazeny
- pokud v CI není dostatek volného místa, vytvoří se nové CI a polovina záznamů z původního CI se přesune do nového - tzv. INTERVAL SPLIT
- pokud není v celé CA volné místo pro nový CI, provede se SPLIT na úrovni CA

Program IDCAMS

S VSAM soubory se pracuje pomocí programu IDCAMS

- programu IDCAMS zadáváme přes SYSIN příkazy k práci s VSAM soubory
- program často vytváří skupinu souborů týkající se jedné databáze
- této skupině se říká klastr

IDCAMS – mazání klastru

```
//DELETEC EXEC PGM=IDCAMS  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
DELETE -  
CTMMSTR.KSDS.CLUSTER -  
CLUSTER -  
PURGE -  
ERASE  
/*
```

- **CLUSTER** – znamená, že se bude mazat cluster, tj. všechny přidružené soubory
- **ERASE** – uložená data se přepíší nulami
- **PURGE** – data se smažou, i když ještě nevypršela jejich platnost

IDCAMS – vytvoření klastru

Nový klastr lze vytvořit takto:

```
DEFINE CLUSTER (NAME ('NEW.VSAM.FILE') -  
    MODEL ('SOME.OTHER.VSAM.FILE') -  
    CYLINDERS (20))
```

Je možné použít následující parametry:

- NONINDEXED - vytvoří VSAM bez indexu tj. ESDS - JCL ekvivalent (DSORG=PS)
- INDEXED - vytvoří VSAM s indexem (KSDS)
- další možnosti jsou LINEAR (LDS), NUMBERED, RELATIVE RECORD (RRDS)
- v případě indexovaných dat se vytvoří dva soubory - jeden pro data, druhý pro index, dohromady jde o tzv. klastr
- index funguje jako slovník, pokud hledáme nějaká dat z/OS se nejprve podívá do indexu, kde najde odkaz do datového souboru

IDCAMS – vytvoření klastru

Indexovaný klastr se vytvoří takto:

```
DEFINE CLUSTER (NAME ('USER.DATABASE') -  
    MODEL ('SOME.OTHER.VSAM.FILE') -  
    CYLINDERS (20) -  
    INDEXED -  
    UNIQUE ) -  
DATA (NAME (USER.DATABASE.DATA)) -  
INDEX (NAME (USER.DATABASE.INDEX))
```

Dále je potřeba udat, na jaké pozici záznamu je uložen klíč a jeho velikost

```
DEFINE CLUSTER (NAME ('USER.DATABASE') -  
    MODEL ('SOME.OTHER.VSAM.FILE') -  
    CYLINDERS (20) -  
    INDEXED -  
    UNIQUE -  
    KEYS ( 20 0) ) -  
DATA (NAME (USER.DATABASE.DATA)) -  
INDEX (NAME (USER.DATABASE.INDEX))
```

Tento příklad říká, že klíč bude začínat na prvním sloupci a bude mít délku 20.

IDCAMS – vytvoření klastru

Velikosti souboru lze udat takto:

```
DEFINE CLUSTER (NAME('USER.DATABASE') -  
  MODEL('SOME.OTHER.VSAM.FILE') -  
  CYLINDERS(20) -  
  INDEXED -  
  UNIQUE -  
  KEYS( 20 0) -  
  RECORDSIZE(500) -  
  RECORDS(2500) ) -  
  DATA(NAME(USER.DATABASE.DATA)) -  
  INDEX(NAME(USER.DATABASE.INDEX))
```

Zde říkáme, že záznamy budou mít velikost 500 bajtů a bude jich 2500.

IDCAMS – vytvoření klastru

Také můžeme specifikovat, na jaký disk se mají soubory uložit:

```
DEFINE CLUSTER (NAME ('USER.DATABASE') -  
    MODEL ('SOME.OTHER.VSAM.FILE') -  
    CYLINDERS (20) -  
    INDEXED -  
    UNIQUE -  
    KEYS ( 20 0) -  
    RECORDSIZE (500) -  
    RECORDS (2500) -  
    VOLUME (DATA1) ) -  
DATA (NAME (USER.DATABASE.DATA) -  
    VOLUME (DATA1) ) -  
INDEX (NAME (USER.DATABASE.INDEX) -  
    VOLUME (DATA2) )
```

IDCAMS – vytvoření klastru

Další možné parametry jsou:

SHAREOPTIONS - definuje sdílení douboru mezi více joby/systémy

- dva parametry - druhý umožňuje sdílení mezi systémy (hodnota 4)
- SHAREOPTIONS(1,3)
 - libovlný počet jobů může číst, pokud chce některý zapisovat, musí mít exkluzivní přístup
- SHAREOPTIONS(2,3)
 - libovolný počet jobů může číst, jeden může i zapisovat aniž by měl exkluzivní přístup
- SHAREOPTIONS(3,3)
 - libovlný job může číst i zapisovat, konzistence není zaručena
- SHAREOPTIONS(4,3)
 - jako 3 s vyšší snahou o konzistenci, ale ta stejně není zaručena

CONTROLINTERVALSIZE - velikost CI **FREESPACE**

- má dva parametry, které udávají, jak velké procento CI a CA má zůstat volné
 - REUSE/NOREUSE
 - udává, zda může být cluster otevřen vícekrát
 - pro vytvoření alternativního indexu je nutné použít NOREUSE
 - UNIQUE/NONUNIQUE
 - zda je k jednomu klíči pouze jeden záznam
 - RECSZ - udává průměrnou a maximální délku záznamu

Pro vložení dat do VSAM klasteru se používá příkaz
REPRO

```
REPRO INFILE(IN) -  
      OUTFILE(OUT)
```

- INFILE, OUTFILE udávají vstupní a výstupní soubor pomocí ddnames
- INDATASET, OUTDATASET - to samé pomocí názvu data setů
- teno příkaz lze použít i pro kopírování dat mezi non-VSAM a VSAM soubory

Lze také vytvořit alternativní index pomocí příkazu DEFINE AIX:

```
DEFINE AIX -  
    (NAME (CTMMSTR.KSDS.NUAIX) -  
    RECORDS (25) -  
    KEYS (20, 12) -  
    NONUNIQUEKEY -  
    RELATE (CTMMSTR.KSDS.CLUSTER) ) -  
DATA -  
    (NAME (CTMMSTR.KSDS.NUAIXDA) ) -  
INDEX -  
    (NAME (CTMMSTR.KSDS.NUAIXIX) )  
DEFINE PATH -  
    (NAME (CTMMSTR.KSDS.NUPATH) -  
    PATHENTRY (CTMMSTR.KSDS.NUAIX) )  
BLDINDEX -  
    INDATASET (CTMMSTR.KSDS.CLUSTER) -  
    OUTDATASET (CTMMSTR.KSDS.NUAIX)
```


Otevření souboru

- funkce `fopen`, `freopen`
- parametry `recfm`, `lreclm` a `blksize` se ignorují
- je nutné udat parametr `type=record`
 - `acc=bwd/fwd` - směr sekvenčního přístupu
 - `passwd` - heslo pro přístup k datům v VSAM

Příklad:

```
f = fopen( "DD:MYVSAM", "r,type=record,acc=fwd" );
```

Nalezení záznamu

- provádí se pomocí funkce `flocate(FILE* cluster, char* key_value, int key_size, int cond)`
- v daném clusteru hledá klíč s zadanou hodnotou
- `cond` může být
 - `__KEY_EQ`
 - `__KEY_FIRST` (první klíč klasteru)
 - `__KEY_GE`
 - `__KEY_LAST`
 - `__KEY_EQ_BWD` (zpětný chod)

Čtení záznamu

- provádí funkce `fread(void* buffer, int num, int rec_size, FILE* f)`
- pokud chceme číst více dat, než má současný záznam, načte se pouze tento záznam a vrátí se počet načtených bajtů
- pokud chceme číst méně dat, než má záznam, načte se jen požadované množství
- opakované volání funkce `fread` znamená sekvenční přístup k záznamům - jeden po druhém
 - KSDS, KSDS AIX, ESDS AIX je posloupnost udána klíči
 - u RRDS je posloupnost udána čísly
 - klíče, pro než neexistují záznamy se přeskakují

Zápis záznamu

- provádí funkce `fwrite(void* buffer, int num, int rec_size, FILE* f)`
- mezi voláním `fread` a `fwrite` se musí volat `fflush` pro zápsaní obsahu bufferů
- fce. vrací počet úspěšně zapsaných bytů

Změna záznamu

- provádí funkce `fupdate(void* buffer, int rec_size, FILE* f)`
- k změně záznamu je nutné provést následující operace
 - nalést požadovaný záznam pomocí `flocate`
 - provést čtení pomocí `fread`
 - provést změny záznamu v bufferu
 - volat `fupdate` - mezi tím se nesmí provádět jiné čtení ani zápis
 - pokud `fupdate` selže, je nutné znovu nastavit správnou pozici pomocí `flocate` a az potom volat další funkce