

# Tomáš Oberhuber

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

## 1 VSAM - přehled

## 2 IDCAMS

# VSAM - přehled

**VSAM** = Virtual Storage Acces Methods

Jde o speciální soubory určené pro databázové aplikace -  
IMS, DB2.

Přidávají určitou organizaci k běžným záznamům.

- 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

# VSAM - přehled

- 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

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

# VSAM - příklad vložení záznamu

## 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
- pokud není místo pro nové CA, vložení selže

Programu IDCAMS zadáváme pomocí SYSIN příkazy k práci s VSAM soubory:

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

## Vytvoření nového souboru

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

## Možné parametry:

- NONINDEXED - vytvoří VSAM bez indexu  $\Rightarrow$  ESDS - v JCL ekvivalent (DSORG=PS)
- INDEXED - vytvoří VSAM s indexem (KSDS)
- LINEAR - LDS,
- NUMBERED
- RELATIVE RECORD - RRDS

V případě indexovaných dat se vytvoří dva soubory - jeden pro data, druhý pro index.

Dohromady jde o tzv. klaster.

Index funguje jako slovník, pokud hledáme nějaká data, z/OS se nejprve podívá do indexu, kde najde odkaz na určité místo v datovém souboru.

```
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.

## Udání velikosti souboru:

```
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))
```

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

Také si můžeme říci, 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) )
```

## Další parametry:

- `SHAREOPTIONS` - definuje sdílení souboru mezi více joby/systémy
- `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

Vložení dat do VSAM klasteru pomocí příkazu REPRO:

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

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

## Vytvoření alternativního indexu 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)
```