

Tomáš Oberhuber

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

Videa na Youtube:

- JCL výklad
- JCL cvičení

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

JCL kombinuje funkce:

- shellového skriptu (jde o skript na spouštění programů)
- makefilů (spouštění úloh může být podmíněné)

Data set pro JCL musí mít následující formát (zpětná kompatibilita s děrnými štítky):

- FB (fixed blocked) - RECORD LENGTH = 80
- nejčastěji se používá PDS
- každý job se skládá z jednotlivých kroků, ve kterých jsou spouštěny externí programy nebo volány procedury
- těchto kroků může být až 255 v jednom jobu

Úvod do JCL

Struktura JCL kódu

Základní příkazy JCL

Definice vstupů a výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění programů v JCL

Procedury v JCL

```
000100 //OBETO80J      JOB (90300000),'OBETO80',
000150 // CLASS=A,REGION=4096K,
000200 // MSGLEVEL=(1,1),MSGCLASS=H,NOTIFY=&SYSUID
000300 //STEP1      EXEC PGM=SORT
000310 //STEPLIB DD DSN=SYS1.SICELINK,DISP=SHR
000320 //  DD DSN=SYS1.SORTLPA,DISP=SHR
000400 //SYSIN      DD *
000500     SORT      FIELDS=(1,75,CH,A)
000600 /*
000700 //SYSOUT     DD SYSOUT=*
000800 //SORTIN     DD *
000900 PLUTO
001000 MARS
          ...
001800 /*
001900 //SORTOUT   DD SYSOUT=*
002000 /*
```

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

```
//*_COMMENTS  
//NAME_FIELD_OPERATION_OPERAND1,OPERAND2_COMMENTS  
//_CONTINUATION  
/*_INSTREAM  
/*  
//
```

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD
Operand DISP
Alokace nového data
setu v JCL
Speciální DD jména
Napořádání data setů

Spouštění
programů v
JCL

Procedury v
JCL

- příkazy JCL se píšou velkými písmeny
- řádek s komentářem začíná `//*`
- řádek s JCL příkazy začíná dvěma lomítky `//` - tzv. ID FIELD
- ihned bez mezery následuje tzv. NAME FIELD - jméno jobu, kroku, definice vstupu/výstupu apod.
- je-li za ID FIELD uvedena mezera, značí to pokračování předchozího řádku
 - navázání na předchozí řádek musí začít před 16 sloupcem

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

- za NAME FIELD následuje operace - OPERATION
- dále jsou operandy operace
 - oddělují se čárkou bez mezery - mezera uvozuje komentář
 - musí končit před 72 sloupcem
 - ten indikoval pokračování na dalším děrném štítku
 - pokud není 72 sloupec prázdný, vyžaduje se pokračování řádku

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD
Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména
Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

- konec jobu značí samotná // za nímiž nenásleduje nic
 - není nutné je uvádět, jako konec JCL se pak bere poslední řádek
 - pokud ale za samotným // následují další řádky, nebudou provedeny
- /* v prvních dvou sloupcích značí datový vstup - **INSTREAM**
 - datový vstup lze poznat podle chybějících //, které určují příkaz JCL '/*' proto není povinné
 - je užitečné, pokud je vstupem JCL kód
 - pokud vstup má také obsahovat /* - například kód v C/C++, lze definovat jiný DELIMITER

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD
Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

```
//INPUT DD DATA,DLM=@@  
/* C/C++ code  
@@
```

Příkaz JOB

JOB definuje nový job/úlohu resp. uvozuje hlavičku úlohy.
Obsahuje následující operandy:

- NOTIFY
- MSGLEVEL
- CLASS
- REGION
- TIME

JOBNAME definuje jméno úlohy.

- píše se před slovo JOB
- udává jméno úlohy (identifikátor)
- délka 1-8 znaků
- často je to ID uživatele plus jedno písmeno
- z/OS neumožňuje současný běh dvou jobů se stejným id

Za jménem úlohy se uvádí klíčové slovo `JOB`.

Příkaz JOB - ACCOUNTING INFORMATION

Za slovo JOB se do závorky píše tzv. ACCOUNTING INFORMATION

- nutné zjistit u administrátora
- bez něj nelze job vůbec spustit - záleží na nastavení systému
- jde o kód přidělovaný různým projektům
 - umožňuje např. i sledovat vytížení systému jednotlivými projekty

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD
Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména
Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

Následuje jméno operátora (autora jobu):

- může být povinné
- nemusí se nutně shodovat s id uživatele, ale často se tak volí
- maximální délka je 20 znaků

Příkaz JOB - parametry

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD
Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména
Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

K následujícím parametrům existují defaultní hodnoty a není tedy nutné je uvádět.

- NOTIFY - komu poslat zprávu o dokončení běhu úlohy, může být jiné než id operátora
- MSGCLASS - určuje, kam směřovat zprávy, které vznikají během zpracování jobu
 - např. TSO, tiskárna apod.

Příkaz JOB - parametry

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

- `MSGLEVEL` - úroveň zpráv - kód + alokační zprávy
 - první parametr
 - 0 = JES statements
 - 1 = JES statements, JCL statements, vypíše vkládané katalogované procedury
 - 2 = JES statements, JCL statements
 - druhý parametr
 - 1 = zprávy od JES, SMS a JCL kód
 - 0 = pouze JCL kód (při abnormal end se vypíše všechno)

Příkaz JOB - parametry

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

- CLASS - určuje, do jaké třídy úlohu zařadit (A, B, Q, K)
 - vlastnosti tříd závisí na systémovém nastavení např.
 - A - úlohy běžící méně než 15 vteřin
 - B - úlohy běžící méně než půl hodiny
 - C - úlohy vyžadující pásky
- REGION - udává paměť potřebnou k úspěšnému zpracování jobu
- TIME - udává maximální dobu běhu jobu
 - TIME=(min,sec)
 - TIME=min

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

EXEC definuje nový krok = spuštění programu / utility nebo volání JCL procedury

- lze spouštět programy
 - `//STEPNAME EXEC PGM=program`
- nebo JCL procedury
 - `//STEPNAME EXEC PROC=procedura`

DD - data definition - slouží k určení vstupů a výstupů

```
//MYDATA DD ...
```

- vytvoří "virtuální" data set se jménem MYDATA tzv. DDNAME
- pod tímto jménem pak program s daným data setem pracuje
 - `fp = fopen("DD:MYDATA", "r");`
- také to lze chápat jako vytvoření symbolického odkazu v Unixu
 - `ln -s indata.dat MYDATA`
 - `fp = fopen("MYDATA", "r");`

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD
Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména
Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

```
//SORTOUT DD SYSOUT=*
```

- vytváříme DD se jménem SORTOUT (program SORT ho používá pro výstup)
- přesměrujeme ho na SYSOUT - "standardní výstup" - stdout
- následuje nastavení SYSOUT, * znamená zacházet s výstupem, jak je uvedeno v hlavičce jobu u MSGCLASS

Operand DSN

DSN - DAT SET NAME - jméno data setu, pro který
"symbolický odkaz" vytváříme

- lze použít název DUMMY např. pro zahození výstupu
 - `//SORTOUT DD DUMMY`

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

Cvičení

- vytvořte PDS `UID.INTRO.JCL` pro JCL skripty
 - tj. použijte formát FB, RECORD LENGTH = 80
- vytvořte v něm člena (složku), EXER01
- napište skript pro seřídění textu s názvy planet pomocí programu SORT

Parametry programu SORT

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

```
SORT FIELDS=(1,5,CH,A)
```

- zadávají se přes SYSIN
- SORT - chceme třídít - jde o jeden z mnoha příkazů pro program SORT
- FIELDS=(1,5,CH,A)
 - chceme třídít podle POLE
 - na PRVNÍ pozici logického záznamu
 - o délce 5 znaků
 - s polem se zachází jako se znaky (CH)
 - třídí se vzestupně (A) - ascending sequence

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

DISP - DISPOSITION - dispozice (stav) data setu na začátku, během zpracování a na konci

```
DISP=(status,normal end, abnormal end)
```

```
DISP=(status,normal end)
```

```
DISP=status
```

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

DISP - DISPOSITION - dispozice (stav) data setu na začátku, během zpracování a na konci

`DISP=(status,normal end, abnormal end)`

`DISP=(status,normal end)`

`DISP=status`

Jde o tzv. poziční parametry (POSITIONAL PARAMETERS)
- význam parametru je dán jeho pořadím.

Poziční parametry v JCL

Černý pes.

DOG=BLACK

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

Černý pes.

```
DOG=BLACK
```

Černý pes s dlouhým ocasem.

```
DOG= (COLOR=BLACK, TAIL=LONG)
```

```
DOG= (BLACK, LONG)
```

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

Černý pes.

```
DOG=BLACK
```

Černý pes s dlouhým ocasem.

```
DOG= (COLOR=BLACK, TAIL=LONG)
```

```
DOG= (BLACK, LONG)
```

Pes s černobílým ocasem.

(TAIL je nyní první poziční parametr.)

```
DOG=TAIL=COLOR= (BLACK, WHITE)
```

Operand DISP

Status může být:

- NEW
 - bude vytvořen nový data set, žádný se stejným jménem nesmí existovat
 - defaultní hodnota, pokud není DISP uvedeno
- OLD
 - data set už existuje a job k němu má exkluzivní přístup (nikdo jiný ho nesmí používat)
 - při zápisu dojde ke smazání všech dat

Úvod do JCL

Struktura JCL
kódu

Základní
příklady JCL

Definice
vstupů a
výstupů

Operace DD
Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména
Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

- SHR
 - data set existuje a může být sdílen mezi více joby (i ostatní joby s ním mohou pracovat)
 - u PDS nezabraní současnému zápisu do stejných členů - u PDSE už ano
 - není dobré provádět v JCL zápis do stejné knihovny kde máme zdrojáky
 - je-li v JCL dispozice SHR, může dojít k současnému zápisu do adresáře a následné ztrátě celého členu
 - viz. také DELETE - smaže celé PDS, ne jen jeden člen
- MOD
 - modifikace - data set existuje nebo bude vytvořen, job k němu má exkluzivní přístup, a pokud do data setu něco zapisuje, připojuje se to na konec
 - nelze používat s PDS - při přidávání dat nedochází k aktualizaci adresáře => připojená data jsou většinou ztracena

Operand DISP

(ab)normal end

- **ABNORMAL END** - nastává pokud při běhu jobu nastane chyba
- **NORMAL END** - job/krok proběhl bez chyby

Možné akce spojené s normal / abnormal end jsou:

- DELETE
 - smazat
 - defaultní pro NEW
- KEEP
 - zachovat ale nezařazovat do katalogu
 - defaultní pro OLD o SHR
- CATLG
 - zachovat a zařadit do katalogu
- UNCATLG
 - zachovat, ale odstranit z katalogu
- PASS
 - co udělat s data setem se řeší až v některém dalším kroku, nelze použít jako třetí parametr

Možné akce spojené s normal / abnormal end jsou:

- DELETE
 - smazat
 - defaultní pro NEW
- KEEP
 - zachovat ale nezařazovat do katalogu
 - defaultní pro OLD o SHR
- CATLG
 - zachovat a zařadit do katalogu
- UNCATLG
 - zachovat, ale odstranit z katalogu
- PASS
 - co udělat s data setem se řeší až v některém dalším kroku, nelze použít jako třetí parametr

U PDS se DISP vztahuje na cele PDS, ne na jednotlivé členy

- DELETE - smaže celé PDS, ne jen zvoleného člena

Alokace nového data setu v JCL

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD
Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména
Napojování data setů

Spuštění
programů v
JCL

Procedury v
JCL

- pokud použijeme DISP=NEW, je nutné udat tzv. DCB (DATA CONTROL BLOCK)
 - jde o parametry souboru – velikost, formát, organizace (PS/PDS)
- lze použít příkaz LIKE=data-set-name - použijí se parametry daný data set
 - funguje pouze s SMS a pro data sety na DASD

Velikost data setu

- velikost se udá pomocí SPACE
- SPACE= (UNIT, (PRIMARY-EXTENTS, SECONDARY-EXTENTS, DIRECTORY-BLOCKS), RLSE)
- UNIT - jednotka pro vyjádření velikosti
 - TRK, CYL, KB, MB, REC
- DIRECTORY-BLOCKS se uvádí pro PDS
- parametr RLSE udává, že nevyužité stopy se po uzavření knihovny uvolní

Příklady:

- $SPACE = (TRK, 10)$
 - PS o velikosti 10 stop
- $SPACE = (TRK, (10, 5))$
 - PS s 10 stopami primárně, 5 stop se bude alokovat při zaplnění
- $SPACE = (CYL, 5)$
 - PS o velikosti 5 cylindru
- $SPACE = (TRK, (10, 5, 8))$
 - PDS s 10 stopami primárně, 5 stop se bude alokovat při zaplnění, adresář o velikosti 8 bloků po 256 bajtech
- $SPACE = (1000, (50000, 10000))$
 - PS s 50000 záznamů primárně, 10000 se bude alokovat při zaplnění, průměrná velikost logického záznamu je 1000 bytů

Organizace data setu

Parametr DSORG - DAT SET ORGANIZATION

- udává typ data setu
- DSORG=PS - physical sequential
- DSORG=PO - partitioned organized
- není nutné, PDS se opět pozná podle nenulové velikosti adresáře

Pro vytvoření PDSE je nutné použít `DSNTYPE=LIBRARY`
Parametr `RECFM` - RECORD FORMAT - F, FB, V, VB, U.

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

Cvičení

- v PDS `UID.INTRO.JCL` vytvořte pomocí příkazu `CREATE` ze složky `EXER01` novou složku `EXER02`
- napište skript pro seřídění textu s názvy planet pomocí programu `SORT`
- vstup načítejte ze souboru `UID.INTRO.TEXTY(PLANETY)`
- výstup uložte do PS souboru `UID.INTRO.SORTOUT`

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

JOBLIB, STEPLIB

- slouží k připojení knihovny s programy
- JOBLIB se uvádí ihned za hlavičku - JOB
 - platí pro celou úlohu
- STEPLIB se uvádí na začátku kroku za EXEC
 - platí pro daný krok
 - je-li uvedeno STEPLIB, JOBLIB se nepoužije

Příklad:

```
//STEPLIB DD DSN=knihovna,DISP=SHR
```

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD
Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména
Napořádání data setů

Spouštění
programů v
JCL

Procedury v
JCL

STEPCAT a JOBCAT

- slouží pro nastavení jiného katalogu

SYSPRINT

- používá PL/I a některé další programy pro chybová hlášení

SYSIN

- používá se pro vstupní data např. zdrojový kód pro překladače
- pokud z/OS narazí na ne-JCL příkaz, přepoše ho na SYSIN

SYSOUT

- standardní výstup

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

```
//DATAIN      DD DISP=OLD,DSN=MY.INPUT1  
//           DD DISP=OLD,DSN=MY.INPUT2
```

- zadané data sety se budou jako vstup zpracovávat jeden po druhém
- v Unixu se zadá více vstupních souborů jako argumenty programu
- ty je ale nutné parsovat, zatímco v z/OS v tom není pro programátora žádný rozdíl

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD
Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména
Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

Cvičení

- skript PDS `UID.INTRO.JCL (EXER02)` upravte tak, aby jako vstup bral soubory
 - `UID.INTRO.TEXTY (PLANETY)`
 - `UID.INTRO.TEXTY (PLANETS)`

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD
Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

**Spouštění
programů v
JCL**

Procedury v
JCL

Programy v JCL spouštíme takto:

```
//stepname EXEC PGM=program-name,  
//    TIME=(min,sec),  
//    REGION=memory,  
//    PARM=parameters,  
//    COND=condition
```

TIME

- udává čas CPU nutný ke zpracování úlohy
- `TIME=1440`, `TIME=NOLIMIT`, `TIME=MAXIMUM`
 - program může běžet libovolně dlouho
- `TIME=0`
 - pokud je to použito spolu s `EXEC` (ne v hlavičce jobu) program může využít všechnen čas CPU, který zbývá z předchozích kroků

REGION

- udává množství paměti potřebné pro úlohu

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

PARM

- předává parametry
- v z/OS se parametry programů většinou předávají pomocí SYSIN
- PARM používají hlavně překladače a linkery/bindery

Spouštění programů v JCL - podmínky

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

COND - vyjádření podmíněného spuštění kroku v závislosti na výsledcích předchozích kroků

- lepší je používat IF-THEN-ELSE
- COND=ONLY
 - tento krok spustit, pouze pokud některý předchozí skončil s ABEND
- COND=EVEN
 - tento krok spustit, i pokud některý z předchozích skončil s ABEND
- COND= (8 , LT , STEP 1)
 - je-li 8 menší než (LESS THAN - LT) návratový kód kroku STEP1, tento krok nebude spuštěn
- COND= (0 , NE)
 - je-li 0 nerovno (NOT EQUAL - NE) maximu návratových kódů předchozích kroků, tento krok nebude spuštěn

Spouštění programů v JCL - návrátové kódy

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD
Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

Návratové kódy

- job vrací maximum návratových kódů jednotlivých kroků
- 0 - vše proběhlo v pořádku
- následují čísla 4,8,12,16
 - čím vyšší, tím závažnější chyba, lze vracet i větší čísla, ale většina programů dodržuje tuto konvenci

Spouštění programů v JCL - návrátové kódy

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

Podmínky pro COND

- LT - less than
- LE - less than or equal to
- EQ - equal to
- NE - not equal to
- GE - greater than or equal to
- GT - greater than

- definice procedury se provede příkazem PROC, který předchází jméno procedury
 - dále lze uvést defaultní hodnoty některých parametrů
 - jméno nesmí být delší, než 8 znaků
- následuje tělo procedury
 - v něm se můžeme odkazovat na jednotlivé parametry pomocí znaku &
 - např. &SRTOUT
- tělo procedury se ukončuje příkazem PEND s mezerou
 - // PEND
- volání procedury tvoří jeden krok úlohy
 - provádí se příkazem
 - //STEPNAME EXEC
PROC=jméno-procedury, PARAM1=param1, PARAM2=param2
 - PROC= může být vynecháno
 - následně můžeme doplnit nebo předefinovat DD použité v proceduře

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

```
//SORTPROC PROC
//MYSORT EXEC PGM=SORT
//STEPLIB DD DSN=SYS1.SICELINK,DISP=SHR
// DD DSN=SYS1.SORTLPA,DISP=SHR
//SORTIN DD DISP=SHR,DSN=&SRTIN
//SORTOUT DD DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(10,5,10)),
//          DSN=&SRTOUT,UNIT=DISK
//SYSOUT DD SYSOUT=*
// PEND
//*****
//STEP1 EXEC PROC=PROC,SRTIN=OBETO80.INTRO.DATA(PLANETY),
//          SRTOUT=OBETO80.INTRO.SORTOUT(PLANETY)
//SYSIN DD *
        SORT FIELDS=(1,3,CH,A)
/*
```

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD

Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména

Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

Cvičení

- vytvořte složku `UID.INTRO.JCL (EXER03)` a volání programu `SORT` obalte do procedury `SORTPROC`

Katalogované procedury v JCL

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD
Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména
Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

- jde o procedury uložené v knihovně
- jméno členu PDS, kam proceduru uložíme musí souhlasit se jménem procedury
 - k definici katalogované procedury se nepřidává hlavička úlohy
- před voláním procedury v JCL je potřeba knihovnu "připojit" pomocí příkazu JCLLIB
 - `//MYLIB JCLLIB ORDER=OBETO80.INTRO.JCLLIB`

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupů a
výstupů

Operace DD
Operand DISP

Alokace nového data
setu v JCL

Speciální DD jména
Napojování data setů

Spouštění
programů v
JCL

Procedury v
JCL

Cvičení

- vytvořte PDS `UID.INTRO.JCLLIB` stejně jako `UID.INTRO.JCL`
- uvnitř vytvořte složku `SORTPROC` a do ní vložte definici procedury `SORTPROC`
- v `UID.INTRO.JCL` vytvořte složku `EXER04` a pomocí katalogované procedury spusťte program `SORT`