

Tomáš Oberhuber

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

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

- 1 Úvod do JCL
- 2 Struktura JCL kódu
- 3 Základní příkazy JCL
- 4 Definice vstupních a výstupních dat - DD
- 5 Alokace nového data setu v JCL
- 6 Parametry programu SORT
- 7 Napojování data setů
- 8 Procedury v JCL
- 9 Příklady

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

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
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

```
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
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

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

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

- příkazy JCL se píšou velkými písmeny
- řádek s komentářem začíná `//*`
- řádek s JCL příkazem začíná dvěma lomítkama `//` - 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
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

- 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
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

- konec jobu značí samotná // za nimiž 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
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

```
//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

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

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
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

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

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

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
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

- `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
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

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

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

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`

Definice vstupních a výstupních dat DD

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

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");`

Definice vstupních a výstupních dat DD

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

```
//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

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

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
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

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
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

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
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

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
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

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)
```

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

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 k smazání všech dat

Parametry pro DD - DISP

Úvod do JCL

Struktura JCL
kódu

Základní
příklady JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

- 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

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

(ab)normal end

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

Parametry pro DD - DISP

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

Parametry pro DD - DISP

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
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

- pokud použijeme DISP=NEW, je nutné udat tzv. DCB (DATA CONTROL BLOCK) parametry
- lze použít příkaz LIKE=data-set-name - použijí se parametry daný data set
 - funguje pouze s SMS a pro data sety na DASD
- 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 stopama primárně, 5 stop se bude alokovat při zaplnění
- `SPACE= (CYL, 5)`
 - PS o velikosti 5 cylindru
- `SPACE= (TRK, (10, 5, 8))`
 - PDS s 10 stopama primárně, 5 stop se bude alokovat při zaplnění, adresář o velikosti 8 traků
- `SPACE= (1000, (50000, 10000))`
 - PS s 50000 záznamů primárně, 10000 se bude alokovat při zaplnění, průměrná velikost logického záznamu je 1000 bytů

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.

Alokace nového data setu v JCL

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

**Alokace
nového data
setu v JCL**

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

```
001_//OBETO80J_JOB_(90300000),'OBETO80',CLASS=A,REGION=4096K,  
002_//_MSGLEVEL=(1,1),MSGCLASS=H,NOTIFY=&SYSUID  
003_//STEP1_ EXEC_PGM= SORT  
004_//STEPLIB_DD_DSN=SYS1.SICELINK,DISP=SHR  
005_//_DD_DSN=SYS1.SORTLPA,DISP=SHR  
006_//SORTIN_DD_DISP=SHR,DSN=OBETO80.INTRO.DATA(PLANETS)  
007_//SORTOUT_DD_DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(10,5,10)),  
008_//_DSN=OBETO80.INTRO.DATA(SORTOUT)  
009_//SYSOUT_DD_SYSOUT=*  
010_//SYSIN_DD_*  
011_ SORT_FIELDS=(1,3,CH,A)  
012_/*
```

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

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
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

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, přepošle ho na SYSIN

SYSOUT

- standardní výstup

Parametry programu SORT

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

```
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
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

**Napojování
data setů**

Procedury v
JCL

Příklady

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

Zadané data sety se budou jako vstup zpracovávat jeden po druhém.

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

```
//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.

Napojování data setů - příklad

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

**Napojování
data setů**

Procedury v
JCL

Příklady

```
001 //OBETO80J JOB (90300000), 'OBETO80', CLASS=A, REGION=4096K,  
002 //          MSGLEVEL=(1,1), MSGCLASS=H, NOTIFY=&SYSUID  
003 //STEP1   EXEC PGM=SORT  
004 //STEPLIB DD DSN=SYS1.SICELINK, DISP=SHR  
005 //      DD DSN=SYS1.SORTLPA, DISP=SHR  
006 //SORTIN DD DISP=SHR, DSN=OBETO80.INTRO.DATA (PLANETS)  
007 //          DD DISP=SHR, DSN=OBETO80.INTRO.DATA (PLANETY)  
007 //SORTOUT DD DISP=(NEW,CATLG,DELETE), SPACE=(TRK,(10,5,10)),  
008 //      DSN=OBETO80.INTRO.DATA (SORTOUT)  
009 //SYSOUT DD SYSOUT=*  
010 //SYSIN DD *  
011   SORT FIELDS=(1,3,CH,A)  
012 /*
```


Spouštění programů v JCL

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

**Napojování
data setů**

Procedury v
JCL

Příklady

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

TIME

- udává čas CPU nutný k proběhnutí 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
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

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

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 , STEP1)
 - 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
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

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
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

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

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedúry v
JCL

Příklady

```
//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)
/*
```

- 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

Katalogované procedury v JCL

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

- 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 nebo pomocí STEPLIB, JOBLIB
 - `//MYLIB JCLLIB ORDER=OBETO80.INTRO.JCL`

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

OBETO80.INTRO.JCL(MYSORT)

```
210 //SORTPROC PROC
300 //MYSORT EXEC PGM=SORT
310 //STEPLIB DD DSN=SYS1.SICELINK,DISP=SHR
320 // DD DSN=SYS1.SORTLPA,DISP=SHR
400 //SORTIN DD DISP=SHR,DSN=&SORTDSN
500 //SORTOUT DD SYSOUT=*
600 //SYSOUT DD SYSOUT=*
610 // PEND
```

Katalogované procedury v JCL

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

```
100 //OBETO80A JOB (90300000), 'OBETO80', CLASS=A, REGION=4096K,  
200 // MSGLEVEL=(1,1), MSGCLASS=H, NOTIFY=&SYSUID, TYPRUN=SCAN  
300 //MYLIB JCLLIB ORDER=OBETO80.INTRO.JCL  
630 //STEP1 EXEC SORTPROC, SORTDSN=OBETO80.INTRO.DATA (PLANETS)  
631 //SYSIN DD *  
632 SORT FIELDS=(1,5,CH,A)  
633 //*STEPLIB DD DSN=OBETO80.INTRO.JCL, DISP=SHR  
660 /*
```

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

1. Kopírování PS data setů - program IEBGENER - příklad COPY1

```
//OBETO80J      JOB (90300000), 'OBETO80', CLASS=A, REGION=4096K,  
//              MSGLEVEL=(1,1), MSGCLASS=H, NOTIFY=&SYSUID  
//COPY         EXEC PGM=IEBGENER  
//*STEPLIB DD DSN=SYS1.SICELINK, DISP=SHR  
//*  DD DSN=SYS1.SORTLPA, DISP=SHR  
//SYSIN       DD DUMMY  
//SYSPRINT DD SYSOUT=*  
//SYSOUT      DD SYSOUT=*  
//SYSUT1     DD DSN=OBETO80.INTRO.SOURCE, DISP=OLD  
//SYSUT2     DD DSN=OBETO80.INTRO.DEST,  
//           DISP=(NEW,CATLG,DELETE),  
//           LIKE=OBETO80.INTRO.SOURCE,  
//           UNIT=DISK
```

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

2. Kopírování PDS (knihoven) - program IEBCOPY - příklad COPY2

```
//OBETO80J      JOB (90300000), 'OBETO80', CLASS=A, REGION=4096K,  
//              MSGLEVEL=(1,1), MSGCLASS=H, NOTIFY=&SYSUID  
//COPY         EXEC PGM=IEBCOPY  
//SYSIN        DD DUMMY  
//SYSPRINT     DD SYSOUT=*  
//SYSOUT       DD SYSOUT=*  
//SYSUT1       DD DSN=OBETO80.INTRO.DATA, DISP=SHR  
//SYSUT2       DD DSN=OBETO80.INTRO.DATA2,  
//              DISP=(NEW,CATLG,DELETE),  
//              LIKE=OBETO80.INTRO.DATA,  
//              UNIT=DISK
```

Pokud na SYSUT2 nastavíme stejný název jako na SYSUT1 dojde ke kompresi knihovny.

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

3. Skript pro výpis PS na SYSOUT pomocí IEBCGENER - příklad PRINT1

```
//OBETO80J      JOB (90300000) , 'OBETO80' , CLASS=A , REGION=4096K ,  
//              MSGLEVEL=(1,1) , MSGCLASS=H , NOTIFY=&SYSUID  
//COPY         EXEC PGM=IEBCOPY  
//SYSIN        DD DUMMY  
//SYSPRINT     DD SYSOUT=*  
//SYSOUT       DD SYSOUT=*  
//SYSUT1       DD DSN=OBETO80.INTRO.DATA , DISP=SHR  
//SYSUT2       DD DSN=OBETO80.INTRO.DATA2 ,  
//              DISP=(NEW,CATLG,DELETE) ,  
//              LIKE=OBETO80.INTRO.DATA ,  
//              UNIT=DISK
```

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

4. Skript pro připsání řádku do data setu pomocí IEBGENER - příklad PRINT2

```
//OBETO80J      JOB (90300000) , 'OBETO80' , CLASS=A , REGION=4096K ,  
//              MSGLEVEL=(1,1) , MSGCLASS=H , NOTIFY=&SYSUID  
//COPY         EXEC PGM=IEBGENER  
//SYSIN        DD DUMMY  
//SYSPRINT     DD SYSOUT=*  
//SYSOUT       DD SYSOUT=*  
//SYSUT1       DD *  
-----  
//SYSUT2       DD DSN=OBETO80.INTRO.SOURCE ,  
//              DISP=(MOD)
```

5. Skript pro spuštění vnořeného jobu pomocí IEBGENER - příklad SUBJCL

```
//OBETO80J      JOB (90300000), 'OBETO80', CLASS=A, REGION=4096K,  
//              MSGLEVEL=(1,1), MSGCLASS=H, NOTIFY=&SYSUID  
//PRINT        EXEC PGM=IEBGENER  
//SYSIN        DD DUMMY  
//SYSPRINT     DD SYSOUT=*  
//SYSOUT       DD SYSOUT=*  
//SYSUT2       DD SYSOUT=(*,INTRDR)  
//SYSUT1       DD DATA  
//OBETO80J      JOB (90300000), 'OBETO80', CLASS=A, REGION=4096K,  
//              MSGLEVEL=(1,1), MSGCLASS=H, NOTIFY=&SYSUID  
//ALLOC        EXEC PGM=IEFBR14  
//JUNKDD       DD DSN=OBETO80.INTRO.JUNK,  
//              DISP=(NEW,CATLG,DELETE),  
//              SPACE=(TRK,(5,1)),  
//              RECFM=FB,  
//              LRECL=80,  
//              BLKSIZE=3120,  
//              UNIT=DISK  
//  
/*
```

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

- příkaz `SYSOUT=(*,INTRDR)` říká, že data se nejprve pošlou programu `INTRDR`, a potom na `*` (zastupuje `MSGCLASS` v hlavičce úlohy)
 - `INTRDR` je `INTERNAL READER`, který načítá JCL kód - viz. část o JES
- `DD DATA` se používá pro vkládání JCL kódu.

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

- program IEFBR14 pouze vrací nulový návratový kód (při běhu nedošlo k chybě) jinak nedělá nic
 - používá se pokud chceme vytvořit nebo smazat určitý data set
 - název pochází z instrukce BR 14 - což odpovídá příkazu RETURN
 - DDNAME při použití IEFBR14 může být libovolné

```
//OGDEN1 JOB 1,BILL,MSGCLASS=X
// EXEC PGM=IEFBR14
//A DD DSN=OGDEN.LIB.CNTL,DISP=(NEW,CATLG),VOL=SER=WORK02,
//      UNIT=3390,SPACE=(CYL,(3,1,25))
//B DD DSN=OGDEN.OLD.DATA,DISP=(OLD,DELETE)
```

Úvod do JCL

Struktura JCL
kódu

Základní
příkazy JCL

Definice
vstupních a
výstupních dat
- DD

Alokace
nového data
setu v JCL

Parametry
programu
SORT

Napojování
data setů

Procedury v
JCL

Příklady

- program pochází od jedné skupiny programátoru v IBM, kteří názvy svých programů uvozovali pomocí IEF
- BR v assembleru znamená skok na adresu
- skok na adresu uloženou v registru 14 způsobí ukončení programu