

Tomáš
Oberhuber

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Moduly

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

Tomáš Oberhuber

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

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Moduly

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

- 1 Programovací jazyky v z/OS
- 2 Vytvoření spustitelného kódu
- 3 Moduly
- 4 Programování v C/C++ v z/OS
- 5 Překladač C/C++
- 6 Linkování programů
- 7 Spuštění programu v TSO/E
- 8 Použití katalogovaných procedur
- 9 Argumenty příkazové řádky
- 10 Práce se soubory

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Modules

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

- **Assembler**
 - umožňuje kontrolu nad kódem na úrovni bitů/bytů
 - vhodný pro psaní subrutin volaných z jiných jazyků
 - z/OS je napsaný v assembleru

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Moduly

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
gováných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

- COBOL - Common Bussines-Oriented Language
- PL/I
- C/C++
- Java
- CLIST
 - obdoba shellových skriptů
 - prakticky se ale nepoužívá, nahrazuje ho Rexx
- Rexx - Restructured Extended Executor

LANGUAGE ENVIRONMENT

- systémové API použitelné ve všech jazycích

Vytvoření spustitelného kódu

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Modules

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

- velké programy se dělí na menší logické celky - moduly
- v C++ většinou jednu třídu implementujeme v jednom souboru
- z jednoho souboru můžeme volat funkce nebo metody implementované někde jinde jejichž kód není známý

Vytvoření spustitelného kódu

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Moduly

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

Pro urychlení vytváření spustitelného kódu se proces překladač dělí na několik kroků:

- Preprocessing/prekompilace
- Kompilace
- Linkování
- Spuštění programu

Preprocessing/prekompilace

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Modules

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

- jde o zpracování maker preprocesoru #include, #define
- při programování se subsystémy CICS (transakce), DB2(databáze) se nepoužívají funkce ale makra
 - EXEC SQL ...
- to je stejné i v Unixu nebo ve Windows

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Moduly

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

- zdrojový kód zpracovaný preprocesorem se předá překladači
- ten vytváří strojový kód tzv. object modul - ten ještě nelze spouštět
- překladač nezná kód ani adresu funkcí implementovaných v jiných modulech
- místo toho si poznamená pouze jméno volané funkce - symbolický odkaz
- nahrazení symbolických odkazů je práce pro linker (binder)

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Moduly

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

- linkování provádí tzv. LINKER, mainframe používá i tzv. BINDER
- binder je "modernější" - umožňuje např. uložit výstup do PDSE
- linker potom bere všechny objektové moduly a nahrazuje symbolické odkazy skoky na skutečné adresy nebo vloží patřičný binární kód
- to, co dělá linker se někdy může provádět za chodu programu, kdy objektový modul linkuje dynamicky - DLL

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Modules

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

- dříve obstarával tzv. BATCH LOADER
- dnes je nahrazen BINDERem
- načítá dané moduly do VIRTUAL STORAGE a spouští je

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Moduly

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

- ukládají se do PDS, tím dostáváme knihovnu
- zdrojové texty - SOURCE (MODUL) LIBRARY
- objektové moduly - OBJECT (MODUL) LIBRARY
- spustitelné programy - LOAD (MODUL) LIBRARY

Programování v C/C++ v z/OS

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Moduly

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

Je několik způsobů, jak překládat a linkovat programy v C/C++

- v závislosti na adresování 16. bitové, 31. bitové a 64. bitové
- s využitím IPA = Interprocedural Analysis
 - překladač běžně provádí jen optimalizace na úrovni jednotlivých procedur
 - pomocí parametru IPA(NOLINK) provede překladač první fázi překladu, kdy do výsledného kódu přidá dodatečné informace
 - druhá fáze se volá s parametrem IPA(LINK), kdy dojde k samotnému provedení optimalizací
- s využitím XPLINK = Extra Performance Linking
 - jde o optimalizaci pro rychlé volání funkcí

Cvičení:

Vytvořte následující PDS:

- `UID.INTRO.C`
 - formát např. FB, RLENGTH = 80
 - v tomto PDS vytvořte složku HELLO a napište program *Hello world!* v C
- `UID.INTRO.OBJ`
 - formát např. FB, RLENGTH = 80
- `UID.INTRO.LOAD`
 - formát U, RLENGTH = 80

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Moduly

Programování
v C/C++ v
z/OS

**Překladač
C/C++**

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

```
//OBETO80J JOB (90300000),'OBETO80',CLASS=A,REGION=48M,  
// MSGLEVEL=(1,1),MSGCLASS=H,NOTIFY=&SYSUID  
// JCLLIB ORDER=(CEE.SCEEPROC,CBC.SCBCPRC)  
//COMPILE EXEC PGM=CCNDRVR,  
// PARM='/SEARCH(''CEE.SCEEH.'') NOOPT SO OBJ LIST'  
//STEPLIB DD DSNAME=CEE.SCEERUN,DISP=SHR  
// DD DSNAME=CEE.SCEERUN2,DISP=SHR  
// DD DSNAME=CBC.SCCNCMP,DISP=SHR  
//SYSLIN DD DSNAME=OBETO80.INTRO.OBJ(HELLO),DISP=SHR  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD DSNAME=OBETO80.INTRO.C(HELLO),DISP=SHR
```

Překladač pro C/C++ se jmenuje CCNDRVR

Parametry se předávají pomocí PARM:

```
// PARM=' [run-time-options]/program-parameters'
```

- zdvojené " se používá pro vložení apostrofu do předávaného řetězce
- SEARCH ("CEE . SCEEH . +")
 - kde jsou uloženy hlavičkové soubory - obdoba -I u gcc
 - LSEARCH se používá pro uživatelské hlavičkové soubory (ne systémové)
- NOOPT - NOOPTIMIZE/OPTIMIZE
 - žádné optimalizace, jiná možnost je OPT(2) nebo OPT(3) - použije se IPA
- SO - SOURCE/NOSOURCE
 - vypisuje zdrojový kód a některé diagnostické zprávy
- OBJ - OBJECT/NOBJECT
 - zda se má vytvářet object modul
- LIST - vypíše výsledný kód v assembleru

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Moduly

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

- **DEBUG** - překlad s informacemi pro ladění
- **ARCH** - architektura
 - 0 - spustitelné všude
 - 5 - na z/900
- **CHECKOUT** - varování před možnými chybami v kódu
- **COMPACT** - zmenšuje výsledný kód
- **DEFINE** - definice pro preprocesor
- **DLL** - generování DLL kódu
- **GOFF** - Generalized Object File Format
- **INLINE** - vkládané funkce
- **IPA** - Interprocedural Analysis

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Moduly

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

- LP64/ILP32 - 64. bitový nebo 31. bitový překlad
- OPTFILE - udává data set s volbami pro optimalizace
- PPOONLY - spustit pouze preprocesor
- RTTI - run-time type information - umožňuje použít `dynamic_cast`
- SQL - pro zpracování SQL maker
- UNROLL - rozbalování smyček
- XPLINK - Extrapformance linking

Použité DDNAMES:

- `SYSIN` - zdrojový kód
- `SYSLIB` - **systemové hlavičkové soubory**
- `SYSOUT` - **chybová hlášení**
- `STEPLIB` - **run-time knihovny**

Překladač C/C++

V JCL pro překlad je nutné připojit následující knihovny:

- CEE . SCEERUN - PDS knihovna obsahující run-time rutiny pro C/C++, COBOL, PL/I, FORTRAN
- CEE . SCEERUN2 - PDSE knihovna obsahující run-time rutiny pro C/C++ a COBOL
- CBC . SCCNCMP -

Cvičení:

- pomocí překladače CCNDRVR přeložte zdrojový kód v `UID.INTRO.C (HELLO)` a výsledný object modul uložte do `UID.INTRO.OBJ (HELLO)`
- příslušný JCL skript uložte v `UID.INTRO.JCL (HELLOCMP)`

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Moduly

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

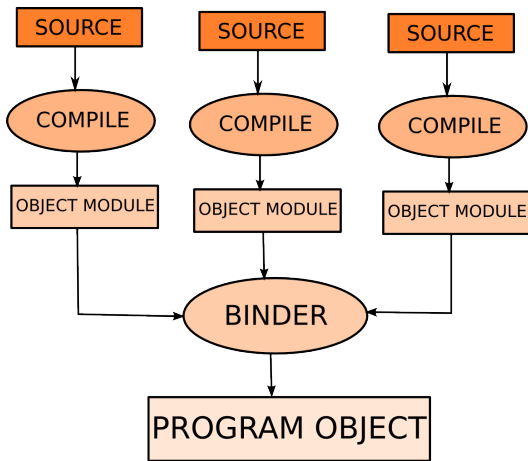
K linkování se používá tzv. **BINDER**.

Existují tři způsoby, jak linkovat program:

- jedno jediné linkování
- linkování jednotlivých objektových modulů
- přelinkovat změněný object modul

Program pro linkování se jmenuje `IEWL`.

Linkování programů - jedno jediné linkování



Programovací jazyky v z/OS

Vytvoření spustitelného kódu

Moduly

Programování v C/C++ v z/OS

Překladač C/C++

Linkování programů

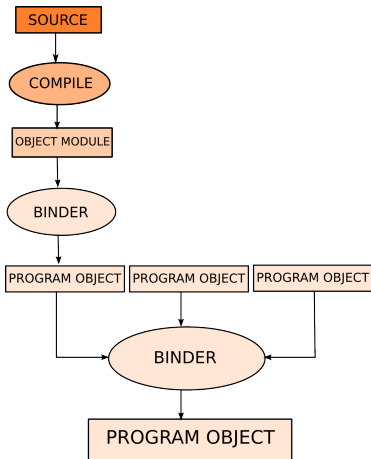
Spuštění programu v TSO/E

Použití katalogovaných procedur

Argumenty příkazové řádky

Práce se soubory

Linkování programů - linkování jednotlivých objektových modulů



Programovací jazyky v z/OS

Vytvoření spustitelného kódu

Modules

Programování v C/C++ v z/OS

Překladač C/C++

Linkování programů

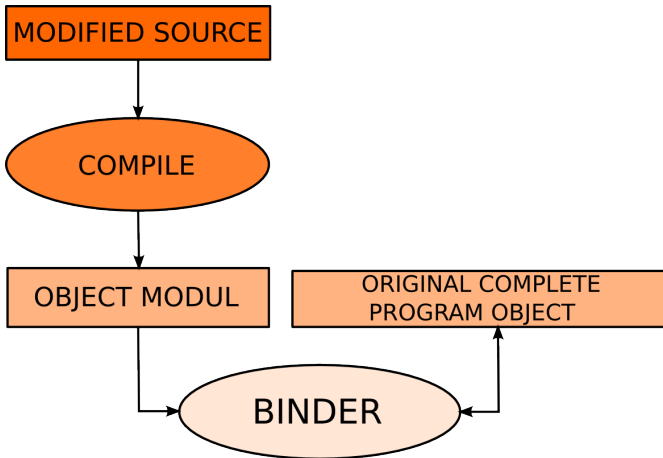
Spuštění programu v TSO/E

Použití katalogovaných procedur

Argumenty příkazové řádky

Práce se soubory

Linkování programů - přelinkovat změněný object modul



Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Moduly

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Moduly

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

Cvičení:

- pomocí linkeru IEWL slinkujte object modul
`UID.INTRO.OBJ (HELLO)` do spustitelného programu
`UID.INTRO.LOAD (HELLO)`
- příslušný JCL skript uložte v
`UID.INTRO.JCL (HELLOBND)`

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Moduly

Programování
v C/C++ v
z/OS

Překladač
C/C++

**Linkování
programů**

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

```
000100 //OBETO80J JOB (90300000), 'OBETO80', CLASS=A, REGION=48M,
000200 // MSGLEVEL=(1,1), MSGCLASS=H, NOTIFY=&SYSUID
000220 //BIND EXEC PGM=IEWL, PARM='OPTIONS=OPTS'
000230 //OPTS DD *
000240 AMODE=31
000250 /*
000260 //SYSLIB DD DISP=SHR, DSN=CEE.SCEELKEX
000270 // DD DISP=SHR, DSN=CEE.SCEELKED
000280 // DD DISP=SHR, DSN=CEE.SCEECPP
000290 //SYSLIN DD DISP=SHR, DSN=OBETO80.INTRO.OBJ(HELLO)
000300 //SYSLMOD DD DISP=SHR, DSN=OBETO80.INTRO.LOAD(HELLO)
000400 //SYSPRINT DD SYSOUT=*
```

Spuštění programu v TSO/E

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Modules

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

**Spuštění
programu v
TSO/E**

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

V TSO/E lze použít příkazy:

```
CALL 'program-data-set-member-name'  
  ['parameters']  
CALL 'obeto80.intro.load(hello)'
```

Použití katalogovaných procedur

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Moduly

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

Jazyk C	jazyk C++	Význam
EDCC	CBCC	překlad 31-bitových a 64-bitových programů
EDCCB	CBCCB	překlad a linkování 31-bitových programů
EDCQCB	CBCQCB	překlad a linkování 64-bitových programů
EDCCBG	CBCCBG	překlad, linkování a spuštění 31-bitového programu
EDCQCBG	CBCQCBG	překlad, linkování a spuštění 64-bitového programu

Další procedury viz.:

- C/C++ User's Guide - str. 297, 298,
- Language Environment Programming Guide - str. 105

Program pro vypsání argumentů příkazové řádky

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Moduly

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

Cvičení:

- vytvořte složku `UID.INTRO.C (ARGS)` a napište do ní kód pro vypsání argumentů z příkazové řádky

```
#include <stdio.h>
int main( int argc, char* argv[] )
{
    int i;
    for( i = 0; i < argc; i ++ )
        printf( "Argument_%d->_%s\n", i, argv[ i ] );
    return 0;
}
```

Program pro vypsání argumentů příkazové řádky

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Moduly

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
gováných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

Cvičení:

- vytvořte složku `UID.INTRO.JCL(ARGS)` a napište do ní kód pro překlad a linkování předchozího kódu pomocí JCL procedury `EDCCB`, výsledný spustitelný program uložte v `UID.INTRO.LOAD(ARGS)`

```
//OBETO80J JOB (90300000), 'OBETO80', CLASS=A,  
// REGION=48M, MSGLEVEL=(1,1), MSGCLASS=H,  
// NOTIFY=&SYSUID  
//COMPPRC EXEC PROC=EDCCB,  
// CPARM=' SO LIST',  
// INFILE='OBETO80.INTRO.C(ARGS)',  
// OUTFILE='OBETO80.INTRO.LOAD(ARGS)', DISP=SHR'
```

Program pro vypsání argumentů příkazové řádky

Programovací
jazyky v z/OS

Vytvoření
spustitelného
kódu

Moduly

Programování
v C/C++ v
z/OS

Překladač
C/C++

Linkování
programů

Spuštění
programu v
TSO/E

Použití katalo-
govaných
procedur

Argumenty
příkazové
řádky

Práce se
soubory

Cvičení:

- vytvořte složku *UID.INTRO.JCL* (ARGSJOB) a napište do ní kód pro spuštění programu *UID.INTRO.LOAD* (ARGS), kterému předáte několik argumentů přes příkazovou řádku

```
//OBETO80J JOB (90300000), 'OBETO80', CLASS=A,  
// REGION=4096K, MSGLEVEL=(1,1), MSGCLASS=H,  
// NOTIFY=&SYSUID  
//JOBLIB DD DSN=OBETO80.INTRO.LOAD, DISP=SHR  
//MYARGS EXEC PGM=ARGS,  
//          PARM='/ARG1 ARG2 ARG3 ARG4'  
//SYSOUT DD SYSOUT=*  
/*
```

Cvičení:

- vytvořte složku `UID.INTRO.C (FILE)` a napište do ní kód pro otevření souboru a vypsaní několika řádek textu

```
000001 #include <stdio.h>
000002
000003 int main( int argc, char* argv[] )
000004 {
000005     FILE* f1;
000006     int i;
000007     printf( "Vytvarim_soubor_intro.cfileps\n" );
000008     f1=fopen( "DD:FILE", "w,recfm=fb,lrecl=20,blksize=3120" );
000009     if( f1 == NULL )
000010     {
000011         printf( "Nemohu_otevrit_soubor_intro.cfileps\n" );
000012         return 16;
000013     }
000014     for( i = 0; i < 100; i ++ )
000015         fprintf( f1, "radek_%d_012345678901234567890\n", i );
000016     close( f1 );
000017     return 0;
000018 }
```


Cvičení:

- vytvořte složku *UID.INTRO.JCL(FILE)* a napište do ní JCL kód pro přeložení, slínkování a spuštění předchozího programu

```
000100 //OBETO80A JOB (90300000), 'OBETO80', CLASS=A, REGION=128M,
000200 // MSGLEVEL=(1,1), MSGCLASS=H, NOTIFY=&SYSUID
000220 //COMPPRC EXEC PROC=EDCCBG,
000240 // INFILE='OBETO80.INTRO.C(FILE)',
000241 // CPARM='SO LIST',
000250 // OUTFILE='OBETO80.INTRO.LOAD(FILE), DISP=SHR'
000260 //GO.FILE DD DSN=OBETO80.INTRO.FILEPS,
000270 //          DISP=(NEW,CATLG,DELETE),
000280 //          SPACE=(TRK,(1,1,0)), UNIT=DISK
```