

Ing. Jaroslav K L E Č K A

ZVT - OKR Ostrava

## Spolupráce počítačů různého typu

### 1. Úvod.

V současné době dochází v naší republice jednak k obměně a jednak k velkému rozšiřování výpočetní techniky ve všech odvětvích našeho národního hospodářství. Není třeba zdůrazňovat, že obměna i rozšiřování by se mělo dít jednotným způsobem, což se také ve skutečnosti praktikuje pod řízením centrálních orgánů dokonce v rámci celé Rady vzájemné hospodářské pomoci a to pomocí Jednotné řady počítačů EC. O účelnosti této praxe není třeba dlouze hovořit, vyplývají z ní však různé problémy a to zejména směrem k výpočetní technice staršího data výroby nebo výpočetní technice, která neodpovídá zásadám, stanoveným pro Jednotnou řadu počítačů RVHP.

Je jisté, že všichni lidé, kteří přišli do styku s výpočetní technikou ví, jak různorodý je park počítačů v ČSR. Prakticky teprve před 3-4 roky došlo ke stanovení zásad dovozu a výroby vyšší výpočetní techniky. Do té doby se dováželo do naší republiky prakticky vše, co se ve světě vyrábělo. Jestliže se tedy nyní tato nejednotná technika, navzájem nekompatibilní, má doplňovat či obměňovat, dochází k zásadnímu problému - z a j i š t ě n í k o m p a t i b i l i t y -

tedy možnosti spolupráce dvou nebo více výpočetních systémů, buďto zcela odlišných nebo alespoň částečně podobných. Problém kompatibility se však objevuje i v rámci počítačů jedné řady, ale o tom později.

Je jisté, že již v okamžiku úvah o koupi nového výpočetního systému je zapotřebí vzít na vědomí, že pokud nebudeme kupovat systém zcela identický s tím, který již máme instalován, dojde po jeho instalaci k nutnosti předávat si mezi jednotlivými systémy nejen programy, ale, a to hlavně, data, která nebudou úplně identická. Tento aspekt je třeba mít stále na zřeteli a v žádné maličkosti na jeho zajišťování nezapomínat, protože by se nám opomenutí některé, byť sebemenší složky, nemuselo vyplatit. Pokud se snažíme tento problém vyřešit zajištěním kompatibility až po instalaci nového systému, může to znamenat zpoždění nebo v některých případech i znemožnění vytvoření nebo převedení některého programového systému pro zpracování dat. Další nebezpečím pak také může být vytvoření tak navhočných prostředků, zajišťujících kompatibilitu, že veškeré další zpracování může být trvale poznamenáno neefektivností zpracování.

Jak jsem již řekl, je nutno tedy zajistit mezi spolupracujícími výpočetními systémy předávání programů a dat. Z tohoto důvodu tedy mluvíme o kompatibility **p r o g r a m o - v é** a **d a t o v é**.

## 2. Kompatibilita programová.

Každý výpočetní systém se od systému druhého liší svojí základní sítí strojových instrukcí a to nejen u systémů zcela odlišných, ale i u systémů téže řady, kde systémy různé úrovně se navzájem liší alespoň vybaveností sítě určitými instrukcemi (nižší typ bude mít vybavenost podstatně nižší než tomu bude u typu vyšší úrovně).

Základ programové kompatibility tedy bude tvořit kompatibilita na úrovni strojového kódu. Takovouto kompatibilitou rozumíme to, že program napsaný ve strojovém kódu nebo ale-

spoň na úrovni nižšího programovacího jazyka (Assembler či jiný symbolický jazyk) může být v nezměněné formě zpracován na tom či onom spolupracujícím počítači. V případě, že může být stejný program zpracován na dvou různých systémech, mluvíme pak o kompatibilitě *o b o u s t r a n n ě*. Jestliže je např. u počítačů jedné řady jeden počítač vybaven omezenější sadou instrukcí než počítač druhý, mluvíme pak o kompatibilitě *s m ě r e m n a h o r u*, neboť na počítači vyššího typu mohou být zpracovány programy napsané bez jakéhokoliv omezení pro počítač nižšího typu, ale naopak to nelze bez omezení zajistit. Pokud chceme zajistit i kompatibilitu *s m ě r e m d o l ě*, musíme vybudovat u nižšího typu počítače celou řadu makroinstrukcí a pomocných modulů, které nahraží jednoduché instrukce vyššího počítače.

Tento problém se vyskytuje i v našem Závodě výpočetní techniky, kde pracují v současné době vedle sebe počítače ICT 1905, EC 1040 a IBM 370/145, z nichž co do instrukční sítě jsou počítače EC 1040 a IBM 370/145 kompatibilní ve směru od EC 1040 k IBM 370/145. Počítač IBM 370/145 má k dispozici širší instrukční síť než počítač EC 1040 a vzhledem k tomu, že je v závodě déle než počítač EC 1040, byl vlastní obecný software, tvořený na bázi Assembleru, založen na celé šíři instrukční sítě počítače IBM 370/145. Jestliže jsme tedy chtěli dosáhnout toho, aby již napsané programy (jedná se nyní pouze o programy napsané v Assembleru) mohly pracovat i na počítači EC 1040, museli jsme, pokud jsme nechtěli všechny programy přepisovat pouze pomocí instrukční sítě počítače EC 1040, vypracovat celou řadu makroinstrukcí, nahrazujících ku příkladu instrukce pro dlouhé přesuny - *MVCL* - či pro srovnávání dlouhých znakových řetězců - *CLCL* - či pro práci s jednotlivými znaky - *ICM* a *STCM*.

Podobný problém se tedy vyskytne všude tam, kde spolupracující výpočetní střediska jednoho odvětví, vybavená ku příkladu výpočetními systémy Jednotné řady EC různé úrovně, při čemž si jednotlivá střediska budou chtít mezi různými

systemy předávat programy.

Pokud chceme zajistit programovou kompatibilitu zmíněného typu u počítačů se zcela odlišnou strukturou základních strojových instrukcí, budeme to vždy zajišťovat za cenu obrovského zvýšení nákladů, které budeme muset investovat do koupě či tvorby zvláštních programů, které budou simulovat na jednom počítači činnost jednotlivých strojových instrukcí počítače druhého. Takovými programům se říká *emulátory*. Takovéto programy však budou vždy pracovat velmi neekonomicky, což si jistě každý, kdo přišel do styku s činností strojových instrukcí, dovede představit.

Jestliže tedy chceme zajistit kompatibilitu programovou u počítačů nekompatibilních co do instrukční sítě, budeme nutně muset opustit myšlenku zajištění této kompatibility na úrovni strojového kódu či nižšího programovacího jazyka a budeme muset začít uvažovat o vyšším programovacím jazyce. Dnes již většina i starších výpočetních systémů, u nás instalovaných je vybavena kompilátory vyšších programovacích jazyků a zejména pro COBOL a FORTRAN IV. Je jisté, že úroveň jednotlivých dodávaných kompilátorů je opět závislá na možnostech a vybavení příslušného systému jednotlivými strojovými instrukcemi, tedy úrovní hardware. Rozdíly v úrovni kompilátorů však již nebývají tak markantní, jako rozdíly mezi základními strojovými kódy, protože základní syntaxe vyššího programovacího jazyka by měla být zachována u všech tvůrců kompilátorů. Některé odlišnosti mezi jednotlivými způsoby zápisu zdrojových výroků vyššího programovacího jazyka pro ten či onen kompilátor pak lze upravit buďto ručním překódováním nebo částečně i tvorbou vlastních opravných programů pro změny kódování, které se pak mohou pomocí těchto opravných programů provádět automaticky.

V našem závodě jsme tento problém vyřešili zavedením jednotného programovacího jazyka, kterým se stal v našem případě jazyk COBOL. Prvním počítačem vyššího typu, který byl v našem závodě instalován, byl počítač ICT 1905. Na tomto po-

čítači jsme po jeho instalaci v roce 1968 začínali psát programy pro zpracování hromadných dat na úrovni symbolického jazyka PLAN, který je jakýmsi druhem Assembleru, bohatě vybaveného makroinstrukcemi pro práci s vnějšími zařízeními včetně vnějších pamětí. Důvodem nepoužití vyššího programovacího jazyka ihned po instalaci počítače, pro zmíněný typ zpracování, byla špatná a nepropracovaná funkce kompilátorů jazyka COBOL, dodávaných výrobcem.

Teprve asi v letech 1970-71, kdy jsme začali uvažovat o koupi dalšího počítače, při čemž již tehdy jsme zhruba věděli, že tímto počítačem nebude počítač od téhož výrobce a že vývoj vybavování středisek počítači povede směrem k bytové orientovaným strojům (ICT 1905 je slovoře orientovaný počítač), jsme začali naše programy kódovat v jazyce COBOL. V té době již také byly firmou ICL dodávány takové verze kompilátorů pro COBOL, že bylo možno k programování v tomto jazyce přistoupit.

Po koupi a instalaci počítače IBM 370/145 jsme pak provedli převedení potřebných programů pomocí práce se zdrojovými knihovnamí na magnetických páscích (ICT) respektive na magnetických discích (IBM) a to z počítače ICT 1905 na počítač IBM 370. Převedení bylo provedeno pomocí vlastního konverzního programu, o němž bude v této přednášce ještě zmínka. Vlastní rozdíly mezi způsobem zápisu programů pro jednotlivé kompilátory byly tak nepatrné, že po převedení byly tyto rozdíly opraveny pomocí ručního kódování a za pomoci firmou IBM dovázaného opravného programu IEBUPDTE. Tento pomocný program provádí opravy zdrojových programů, umístěných na magnetickém disku v knihovně zdrojových programů. Při takovémto způsobu převedení se vyskytla potřeba překódovat pouze asi 10% zdrojových výroků.

Již rok po instalaci počítače IBM 370/145 jsme věděli, že dalším počítačem, nahrazujícím počítač ICT 1905 bude některý z počítačů Jednotné řady počítačů ze země RVHP, a to nejpravděpodobněji počítač EC 1040, který by měl být kompa-

tibilní s počítačem IBM 370/145 a to na úrovni operačních systémů OS/MFT či MVT. Z těchto důvodů jsme se rozhodli, že nebudeme pronajímat nové kompilátory pro jazyk COBOL, které nám firma IBM nabízela pro počítač řady 370 a že zůstaneme u verzi kompilátorů, používaných pro operační systém OS/MFT, při čemž vývoj těchto verzí byl právě pro OS/MFT zastaven i s ukončením vývoje operačního systému MFT u verze 21.8. V současné době je operační systém OS/MFT pro počítač EC 1040 kompatibilní s verzí IBM OS/MFT 20.0 (podle pramenů fy ROBOTRON). Znamená to tedy, že z tohoto hlediska bychom měli mít programy mezi jednotlivými počítači kompatibilní.

### 3. Kompatibilita datová.

Datovou kompatibilitu můžeme zhruba rozdělit na dva logické celky.

První logický celek tvoří kompatibilita vstupních a výstupních papírových médií. Každý výpočetní systém je vybaven základní programovou podporou pro zpracování papírových médií. Většinou bývá pro každý systém určen konkrétní, standardně používaný kód papírových médií (děrných štítků a děrných pásek). Pomocí základního instrukčního vybavení pro práci s pomalými periferiemi lze zajistit čtení či děrování papírového média v jakémkoliv kódu či libovolné paritě. Některé operační systémy počítačů III. generace dokonce umožňují automatickou konverzi vnějších kódů do vnitřního kódu, v němž jsou jednotlivé znaky interpretovány ve vnitřní paměti.

Při použití dvou zcela nejednotných systémů však dochází často k tomu, že kódy papírových médií standardně používané u těchto systémů jsou nejednotné. V tomto případě jsou dvě možnosti:

- a) vybavit oba systémy čtecími či děrovacími programy nebo moduly pro zpracování obou rozdílných kódů;
- b) standardizovat používané kódy a na počítači, pro něž

nejsou tyto kódy standardní vytvořit vlastní program či modul pro zpracování takovýchto nestandardních kódů.

První případ je sice realizovatelný, ale neobvyklý i těžko použitelný, vzhledem k organizačním těžkostem při určení, zda se má zpracovávat ten či onen kód a který program či modul bude pro zpracování použit.

V druhém případě je také výhodné, když počet použitých kódů je minimalizován a není nutně vytvářet více standardních zpracovatelských modulů či programů neboť pak se zcela samozřejmě převádí tento případ na případ první.

Tak jako u programové kompatibility, uvedu způsob, jakým jsme řešili tento problém, to znamená zajištění datové kompatibility papírových médií, v Závodě výpočetní techniky OKR.

Počítač ICT 1905 používá pro papírová vstupní i výstupní média kód ISO a to jak pro děrné štítky, tak pro 8 stopou děrnou pásku. Z tohoto důvodu byl také veškerý strojový park, pořizující tato papírová média zapojen tak, aby děroval jak v centru (ve vlastním závodě), tak i v decentru (na jednotlivých podnicích po celém OKR) tento ISO kód. Pro zpracování několika programových systémů je také používána 5 stopá děrná páska v kódu CCITT, která je používána pro přenos dat pomocí dálkopisné sítě. Pro čtení ISO kódu, ale i pro jeho děrování, je počítač ICL 1905 vybaven standardní softwarovou podporou. Pro zpracování kódu CCITT jsme si pak museli udělat na základě základních strojových instrukcí pro práci se znakovými perifériemi vlastní zpracovatelské moduly.

V době příchodu počítače IBM 370/145 jsme tedy měli k dispozici mohutný strojový park (zhruba 150 děrovačů děrné pásky, rozložených po celém revíru) pro pořizování vstupních dat v daném kódu a tomuto kódu jsme se zejména u děrné pásky museli přizpůsobit.

Pro čtení děrných štítků jsme pak zvolili kompromis. Vzhledem k tomu, že ISO kód a IBM kód pro děrné štítky se navzájem liší pouze v některých speciálních značkách, používa-

ných prakticky pouze pro děrování programů, vybavili jsme náš provoz pořizením vstupních dat třemi popisujícími děrovači fy IBM, pracujícími pochopitelně v kódu IBM a dále jsme překódovali jeden ICL přezkoušeč děrných štítků do tohoto IBM kódu. Jelikož data, vstupující do zpracování systémem zpracování hromadných dat, jsou z 95% pouze numerické povahy, nedochází při zpracování děrných štítků na obou počítačích k obtížím. Pouze při snímání alfanumerických údajů docházelo, zejména zpočátku, při děrování programů psaných v Cobolu, k omylům a pro tento případ jsme si vytvořili jednoduchý program, který může na základě parametrů předděrovávat štítky z jednoho kódu do druhého. Tento program pracuje na počítači IBM a to z důvodu, že tento počítač je vybaven výstupním děrovačem děrných štítků.

Pro zpracování děrných pásek není operační systém IBM OS/MT ani OS/VS vybaven prakticky žádnou firemní softwareovou podporou. Současně ani hardware není standardně vybaven snímači DP a naše konfigurace byla dodatečně vybavena československými snímači FS 1501, připojenými na centrální jednotku pomocí nestandardní řídicí jednotky. Z již uvedených důvodů jsme tedy museli přistoupit k vypracování vlastních čtecích a konverzních modulů, použitelných ve všech programovacích jazycích. Máme tedy vypracovány čtecí a konverzní moduly pro použití v jazycích Assembler, Cobol, Fortran IV a PL/1, při čemž všechny moduly jsou stavěny tak, abychom je po menších úpravách mohli použít pro čtení libovolného 8 stopého děrnopáskového kódu. Podobný čtecí modul pak máme pochopitelně vypracován pro zpracování děrné pásky v kódu GCITT. Lze tedy říci, že jednotností používaného strojového parku, co do jednotnosti použitého kódu, jsme měli značně usnadněnou práci při přípravě zpracovatelských modulů, protože jich nebylo nutné tvořit větší množství.

Pro počítač EC 1040 pak budeme používat již existující standardní moduly, používané na počítači IBM 370/145 s tím, že pokud tyto moduly používají zvláštní instrukce ze sítě in-



strukci počítače IBM 370/145, musí se tyto moduly překompilovat za použití makroinstrukcí, o nichž byla zmínka ve druhé kapitole této přednášky.

Druhý a nejdůležitější logický celek pak tvoří datová kompatibilita na úrovni vnějších magnetických paměťových médií. Kompatibilita na této úrovni je velmi silně závislá na hardware instalovaných výpočetních systémů. Velkou roli zde hraje organizace vnitřní paměti a zejména pak struktura základní informace, se kterou se pracuje v centrální jednotce systému. Touto strukturou informace zde rozumíme nejmenší adresovatelnou jednotku (byte - slovo), se kterou můžeme pracovat a dále je pak důležitý rozsah této jednotky v bitech.

Datová kompatibilita na úrovni vnějších paměťových magnetických médií lze opět zjednodušeně rozdělit do dvou částí:

- a) datová kompatibilita systémů se stejnou organizací vnitřní paměti (zobrazení informace);
- b) datová kompatibilita systémů s různou organizací vnitřní paměti.

První a jednodušší případ nastává tehdy, když máme převádět data mezi počítači, pracujícími se stejnou strukturou dat. Pak nám totiž stačí, aby oba systémy byly vybaveny vnějšími paměťovými zařízeními, schopnými zpracovávat stejná magnetická média a to i co do hustoty záznamu. Oba výpočetní systémy by také měly být vybaveny stejným nebo alespoň podobným operačním systémem.

Zde se na chvíli zastavme. Zmínka o shodnosti nebo alespoň podobnosti operačních systémů je důležitá zejména z toho důvodu, že u takovýchto operačních systémů jsou shodně zaznamenávány zejména pomocné informace, jakými jsou různé jmenovky (labels) nosičů. V opačném případě je nutné zpracovávat jmenovky vytvořené pod jiným operačním systémem (vztah mezi DOS a OS) nestandardním způsobem, pomocí zvláštních zpracovatelských modulů, které nabývají součástí operačního systému, a které mají zpracovávat tyto nestandardní jmenovky.

Rozdíl ve jmenovkách nosičů, ale i v práci s jednotlivými datovými soubory je obzvláště markantní u magnetických disků, kdy prakticky nelze tyto nosiče zpracovávat pod jiným operačním systémem bez převodu pomocí magnetických pásek. Jediným operačním systémem, který tuto práci umožňuje, je operační systém pro počítače IBM řady 370, který se jmenuje OS 370/VM (Virtual Machines), který však není v ČSER k dispozici ani na žádném počítači této řady.

U magnetických pásek je možné zpracování jmenovek pomocí modulů operačního systému přeskóčit a zpracovávat pak pouze vlastní uživatelská data.

V případě ZVT se jedná pouze o kompatibilitu mezi počítači IBM 370/145 a EC 1040, kde jsou použity operační systémy OS/VSI (Virtual Storage) respektive OS/MFT. OS/VSI je užíván u počítače IBM a je obdobou OS/MFT, používaného u počítače EC 1040. Datová kompatibilita mezi oběma počítači je zajištěna pouze pomocí magnetických pásek, při čemž u počítače IBM 370/145 budou instalovány dva MP stojany, pracující s dvojnásobnou hustotou záznamu a to 800 respektive 1 600 Bpi. Hustota záznamu je pak přepínatelná pomocí řídicích výroků pro připojení programů k operačnímu systému (JCL), vzhledem k tomu, že počítač EC 1040 je vybaven pouze stojany, které mohou zapisovat a číst maximálně v hustotě 800 Bpi, musí se při pořizování dat, která mají být předána z počítače IBM 370/145 na EC 1040 na tuto potřebu pamatovat a zadat pro přepnutí do požadované hustoty odpovídající parametr. Při převodu dat z počítače EC 1040 na IBM 370/145 se pak přepnutí děje automaticky, protože operační systém OS/VSI si převezme údaj o hustotě záznamu přímo ze jmenovky nosiče. Rozšíření datové kompatibility se pak připravuje pro příští léta tím, že k počítači EC 1040 budou připojeny velkokapacitní disky, kompatibilní s disky IBM 2319.

Nyní si řekněme něco o druhém případě datové kompatibility na úrovni vnějších magnetických médií a sice o datové kompatibilitě dvou výpočetních systémů s n e s t e j n ě

o r g a n i z o v a n o u   v n i t ř n í   p a m ě t í .

Tento případ je z právě přednášeného tématu skutečně problémem největším. Co mne k tomuto tvrzení vede? Na první pohled se zdá, že pro každého zkušeného programátora je velmi jednoduché vypracovat konverzní program, který bude přetvářet data z jednoho druhu organizace do druhého. Pokud bychom chápali problém kompatibility dvou systémů jen tohoto pohledu, pak by to byl problém skutečně triviální. Mohu zde odpovědně říci, že pokud si budeme předávat mezi dvěma různě organizovanými systémy malé množství dat, pak se skutečně můžeme omezit na velmi jednoduchý způsob převádění těchto dat. Zmíněné množství dat, které rozhoduje o složitosti konverzního programu, není ani tak určeno množstvím těchto dat, ale spíše rychlostí příslušné vnější paměti a centrální jednotky počítače, na němž konverze bude probíhat. Z toho vyplývá, že rozhodujícím činitelem je čas, po který bude konverze probíhat. Pokud je tento čas relativně malý pro to, aby se konverzní běh mohl i několikrát opakovat bez velkých ztrát strojového času, nemusíme program příliš zabezpečovat proti různým poruchám.

Jestliže se ale začne množství předávaných dat zvětšovat do té míry, že se časy konverzí začnou počítat na desítky ba i na stovky minut, musíme zde začít uvažovat v docela jiných měřítcích a takováto konverze se vlastně stává zcela samostatným subsystémem pro zpracování hromadných dat, který je pochopitelně nutně zabezpečit proti všem rušivým vlivům. Celý konverzní systém musíme především zabezpečit proti ztrátě jednotlivých fyzických záznamů při zpracování na obou zúčastněných systémech. Znamená to tedy, že do vznikajícího souboru, který bude dále zpracováván, musíme vložit takové pomocné informace, které umožní zpracování určitého úseku dat (ku příkladu po jedné cívice magnetické pásky) provedení kontroly počtu zpracovaných fyzických záznamů - bloků.

Velmi nutné je také zajištění možnosti restartu konverzního programu od určitého místa, bez nutnosti opakovat celý běh od počátku a to vždy v případě jakékoli poruchy chodu

této konverzce sč již z vnějšku či vnitřku. Na příklad firma ICL doporučuje sabudovat do každého programu, pracujícího déle než 20 minut, kontrolní body, od nichž je možné provádět restarty.

Nyní, alespoň ve stručnosti popíši způsob, jakým jsme tuto oblast řešili v našem závodě. Již před příchodem počítače IBM 370/145 jsme potřebovali zajistit pro potřeby ladění systémů, dále zpracovávaných na zmíněném počítači, převedení některých datových souborů ze slovové organizace počítače ICT 1905 do organizace bytové počítače IBM 370/145. Zde je třeba se zmínit, že ve slovové organizaci počítače ICT 1905 lze data zapisovat ve formě 6 bitových znaků, které jsou však vždy zpracovávány či zapisovány ve čtveřicích, tedy vždy do jednoho slova. Dále pak existuje zápis ve formě binárních čísel, uložených v jednom slově - 24 bitů - nebo jednom dvojslově - 48 bitů - nebo ve formě snížených binárních čísel zapisovaných opět do jednoho slova či jednoho dvojslova. V bytové organizaci se pak používá zobrazení ve formě 8 bitových znakových řetězců, 16, respektive 32 či 64 bitových binárních čísel a dále pak ve zhuštěné dekadické formě v délce 1 až 8 bytů (1 až 15 místné dekadické číslo). Pro jednoduchost jsme nevažovali o převodu čísel ve formě pohyblivé řádové čárky, vzhledem k tomu, že toto zobrazení se většinou při hromadném zpracování dat nepoužívá.

Je celkem jasné, že napsat obecný program pro převedení všech forem zápisu informací z jednoho uvedeného typu organizace do typu druhého není úplně jednoduché, obzvláště ne na počítači, který nemá ve své instrukční síti instrukce pro jednoduchou práci se znakovými řetězci a hlavně pak s informací jinak zobrazenou, než s jakou se na daném typu počítače pracuje. U ICT 1905 je práce se znakovými řetězci velmi obtížná a proto i převod ze slovové organizace do bytové formy byl velmi neefektivní.

Uvedl jsem tedy, že v první fázi jsme vytvořili konverzní program na počítači ICT 1905 a teprve později na poči-

tačí IBM a to prostřednictvím sedmistopých stojanů, které jsou instalovány na obou počítačích. Konverzní program na počítači IBM 370/145, provádějící převod z organizace slovové do bytové, pracuje vzhledem k vyhazení počítače instrukcemi pro snadnou práci se znakovými řetězci, ale také vzhledem k vyšší rychlosti centrální jednotky počítače až 100 krát rychleji. Rychlost konverze zde nemůžeme měřit na základě rychlosti přenosu sedmistopých stojanů, ale časem CPU (centrální jednotky), který byl zapotřebí pro provedení konverze, protože tato spotřeba je pak podmínkou pro možnost efektivního multiprogramování na počítači, který konverzi provádí.

V současné době máme zabezpečenou oboustrannou konverzi dat mezi oběma počítači. Konverzi každého jednotlivého souboru dat z počítače ICT musíme vždy samostatně naprogramovat pomocí určitých standardních makroinstrukcí a pevných částí programu a to v jazyce Assembler, při čemž pro každý typ konverze jednoho zobrazení do druhého je samostatný typ makroinstrukce. Standardní pevné části jsou ve zdrojové formě vkopírovány do programu. Program pak pracuje velmi efektivně, protože je vlastně šit na míru konkrétního převáděného souboru dat.

V obráceném směru, tedy od IBM 370/145 na ICT 1905 se každá konverze programuje za pomoci konverzního modulu, vyvolávaného z jakéhokoliv programovacího jazyka, používaného v našem závodě. Činnost tohoto konverzního modulu pro daný typ logického záznamu je řízena pomocí parametrů, které mají standardní formu pro téměř všechny standardní moduly napsané v našem závodě.

Pro vaši představu uvádím, že každý měsíc se v obou směrech převádí rutinně 30-40 datových souborů, z nichž nejméně 10 je vícecívkových s dobou trvání od jedné do tří hodin.

Největším problémem je zpracování standardních IBM jmenovek na počítači ICT. Tyto zmíněné jmenovky jsou totiž konstantně zapisovány v sudé paritě, při čemž vlastní data jsou

pak povinně pro účely konverze zapisována v paritě liché, protože v sudé paritě nelze zapsat znak s binární hodnotou 0. Dále nelze na počítači IBM vytvořit standardní ICT jmenovku, protože tato jmenovka obsahuje Tape Mark, kterýžto znak nelze na počítači zapsat v požadované formě, vzhledem ke konstrukci hardware. Z této nutnosti vyplývá, že pro čtení dat ze souboru pořízeného na počítači IBM pomocí konverzního programu již ve standardní formě, co do tvaru logických i fyzických datových záznamů, je zapotřebí použít vlastní čtecí modul, nevyužívající standardní softwarové podpory, ale pracující se základními strojovými instrukcemi pro práci s periferními zařízeními. Pro zabezpečení správnosti přenosu se při vytváření souboru počítači zapsané fyzické záznamy a výsledná suma těchto záznamů - bloků - je zapsána do zvláštního logického záznamu, který je pak zkoumán čtecím modulem na počítači ICT, který naopak srovná tento počet s počtem přečtených bloků. Pro zabezpečení restartovatelnosti se snažíme soubory členit logicky tak, aby bylo možné započít s konverzí od poslední cívky, která byla skonvertována správně.

#### 4. Závěr.

Závěrem bych chtěl připomenout ještě možnost spojení dvou různých systémů buďto pomocí telekomunikačních cest nebo přímo pomocí sdílených diskových jednotek či přímo propojených centrálních jednotek. Tato možnost však je u nás velmi řídko používána a je poněkud stranou od předneseného tématu a jistě by si vyžádala samostatnou přednášku.