

ÚČELOVÁ VÝUKA PROGRAMOVÁNÍ NA VĚDECKO-VÝZKUMNÉM PRACOVÍŠTI

Samočinné počítače se v současné době využívají v praktických aplikacích zhruba již dvě desetky let. Jak už bylo konstatováno na těchto seminářích v minulých letech, je to dost dlouhá doba k tomu, že se programování stalo řemeslem a normálním zaměstnáním pro mnoho pracovníků a není už jenom výsadou nadšenců, kteří se zájmu o věc jsou ochotni překonávat skoro jakékoliv překážky. I školství vzalo již nástup výpočetní techniky v úvahu a zařadilo do výuky na mnoha školách přednášky na téma "programování samočinných počítačů", či "využití výpočetní techniky" a podobně, čímž se své úlohy z hlediska výchovy nových pracovníků zhostilo. Na pracovištích "v praxi" je však situace trochu jiná. Doba, po kterou uvedené vyučovací přednášky existují, je ještě příliš krátká, než aby se všeobecná znalost výpočetní techniky rozšířila generační výměnou. V závislosti na způsobu a rychlosti, jakou se využívání výpočetní techniky v jednotlivých odvětvích rozšiřuje, tíží příslušná pracoviště více, či méně problém nezpracovanosti jejich některých pracovníků do problematiky využívání výpočetní techniky. Tento příspěvek má uvést hlavní poznatky z jednoho vědeckovýzkumného pracoviště, které se pokusilo vyřešit si otázku dostatečného zpracování svých technických kádrů do problematiky využívání výpočetní techniky vlastními silami.

Jaké jsou podmínky pro takovou akci na pracovišti v praxi? Zatím co obecná školská příprava se zaměřuje (a je to zřejmě správné) více na přehledovost a encyklopedičnost probírané látky a podle oboru, pro nějž vyučuje, na obecnější teoretický základ vyučované disciplíny, je na praktickém pracovišti třeba vzít v úvahu určité výchozí skutečnosti.

První z nich je účelovost výuky. Je zde míněno to, že cílem je naučit vybrané pracovníky programovat a to v daném jazyce, s využitím daného počítače, překladáče a operačního systému a to se vším, co je třeba od počátku přípravy programu, přes jeho ovláčení, až po výpočet se správnými výsledky v prakticky použitelné formě. Teoretické poznatky není nutno podceňovat, avšak nejsou zde prvořadé.

Druhou skutečností je nástavbový charakter výuky. Výuka má charakter dalšího, jakéhosi postgraduálního vzdělávání při zaměstnání, které je sice pro hlavní obor působnosti účastníků účelné, avšak přímo se ho netýká a probíhá bez přerušování, případně jen s malým narušením plnění normálních úkolů.

Třetí skutečností, které je třeba výuku přizpůsobit, je okruh účastníků, kteří jsou vybráni. Rámcově záměrem vedení pracoviště zde bylo, aby přípravou prošla většina středních i vyšších technických kádrů z laboratoří. Dvacetiletá zkušenost z využívání počítačů při řešení výzkumných úkolů na pracovišti totiž dostatečně potvrdila, že při složitějších způsobu využití počítače (simulace procesů, či optimalizace hledaných řešení) a při klasické dělbě práce mezi výzkumníka na jedné straně a matematika-analytika, či programátora na druhé straně je zpravidla dobrá komunikace mezi nimi závažným problémem. Opakovaně se ukázalo, že spojení obou činností (nebo alespoň odborností) v jedné osobě přináší zisk jak ve zjednodušení této složité mezilidské komunikace, tak v parametrech programů, které pak obvykle značně překonávají parametry programů vzniklých při klasické bariéře dělby práce výzkumník - pracovník výpočetní techniky. Odtud vychází záměr vyškolit všechny potenciální budoucí uživatele výpočetní

techniky, kterým se tak připravuje jedna z nutných podmínek pro zvládnutí rozsáhlé problematiky využití výpočetní techniky ve vědeckovýzkumné práci. I když tento cíl lze považovat za ušlechtilý, je třeba počítat, že kromě účastníků s vlastním aktivním, či pasivním zájmem bude vybrána i skupina těch, kteří o účelnosti výuky, do níž jsou zařazeni, budou mít pochybnosti.

Po zvážení celé situace byly stanoveny tyto hlavní zásady pro organizování kurzu:

1. Výuku zaměřit na FORTRAN, který je pro oblast vědeckotechnických výpočtů vhodný a na pracovišti je již mnoho let užíván. Konkrétně se omezit na jeho implementaci na počítač, který má pracoviště k dispozici.

2. Kromě jazyka probrat příkazy pro ovládnutí překladačového programu, pro zadání výpočtu a další potřebné manipulace se soubory. Vysvětlit význam všech sdělení v protokolech o překladu a o nalezených chybách. Probrat způsoby a možnosti provádění oprav v programech.

3. Výuku zajistit kurzem, nevolit na př. samostatné studium, které má na účastníky větší nároky.

4. Periodicitu lekcí zvolit cca 1x týdně, což je kompromis mezi mírou zatížení a mírou zapomínání do příští lekce.

5. Velikost skupiny omezit tak, aby byl umožněn dostatečný rozsah aktivní účasti jednotlivců (max. cca 25 lidí).

6. Osnovu kurzu přizpůsobit tomu, aby účastníci mohli co nejdříve sestavovat vlastní jednoduché, avšak úplné programy.

7. Účastníkům zajistit možnost průběžně ladit své programy na počítači a provádět cvičné výpočty. Zásadně tuto možnost neomezit na jeden program pro účastníka a na jeden den.

8. Při výběru lektorů přihlížet nejenom k jejich pedagogickým schopnostem a znalostem výpočetní techniky, ale i k jejich kvalifikaci v oboru působnosti účastníků kurzu; to podpoří jejich přirozenou důvěru ve smysl kurzu a umožní to volbu lepších cvičných příkladů a účelné konzultace pro budoucí činnost účastníků.

Podle těchto zásad byl uspořádán kurz a byla pro něj stanovena osnova, která byla podle situace v kurzu ještě průběžně zpravená. Lůraz byl kladen na 6. zásadu, již byla přizpůsobena první část kurzu. Cíle bylo dosaženo v 9. lekci úplným praktickým příkladem. Druhá část kurzu navazovala bez přeručení na první a byla zhruba stejně rozsáhlá. Byly v ní probány zbývající prvky jazyka FORTRAN stejnou formou v pedagogicky účelném pořadí. Současné účastníci průběžně pracovali na svých programech.

Při hodnocení výsledků kurzu se ukázala účelnost zásad 6 a 7. Asi třetina účastníků vypracovala jeden nebo několik vlastních programů většího rozsahu. Jejich zaměření předčilo původní očekávání; kromě několika výjimek tyto programy řešily praktické úlohy a jejich výsledky byly prakticky využity. Několik absolventů kurzu využilo výpočetního střediska od té doby již opakovaně.

Předpokládaný rozsah probírané látky se ukázal být zcela dostatečným. Ke konci kurzu, kdy se účastníci již většinou zařadili mezi úspěšné, nebo neúspěšné (středně úspěšných bylo málo), začala klesat účast. U úspěšných proto, že již necítili potřebu dalších informací pro svoji programátorskou činnost, u neúspěšných se neopak stupňoval pocit nepochopení látky, nebo se projevil nedostatek vlastního zájmu.

Jako pokračování této výuky se pro zájemce (každý podle vlastního uvážení) předpokládají nepravidelně pořádané, tématicky orientované krátké semináře, v nichž různí specialisté (vnitřní i externí) sdělí své zkušenosti s využíváním výpočetní techniky v některých, pro pracoviště zajímavých oblastech. (Řešení rovnic, operace s maticemi, numerická integrace, problém numerických chyb, nenumerické a logické úlohy a j.)

Vzhledem k tomu, že výběr látky, zejména pro první část kurzu může být zajímavý i pro jiná pracoviště, je uveden v závěru příspěvku v chronologickém pořadí, včetně rozdělení do lekcí. Hvězdičkou *) jsou označeny probírané nestandardní prvky jazyka, které jsou v implementaci k dispozici.

1. Algoritmický jazyk jako programovací prostředek. Jeho implementace na počítač. Překladač. Zárojková a přeložená verze programu. Řízení počítače operačním systémem.

2. Soubor znaků používaných ve FORTRANu. Identifikátor a jeho syntax. Proměnná. Typ INTEGER, REAL, LOGICAL. Popis typu příkazem a implicitně, písmenová konvence. Konstanty typu INTEGER, REAL a LOGICAL a jejich syntax.

3. Příkazovací příkaz. Aritmetický výraz. Aritmetické operace a jejich priority. Závorky a jejich použití pro změnu implicitní priority. Logický výraz. logické operace a jejich priority. Indexované proměnné. Rozsah indexů. Rozměrová specifikace v popisu typu. Indexový výraz. Příklady na aritmetické a logické výrazy.

4. Relační operátory a relační výrazy. Rozšíření priority na relační operace. Příkaz GO TO. Aritmetické a logické IF. Příklad na větvení programu pomocí IF. Cykl pomocí IF.

5. Příklady programů pro jednoduché matematické úlohy (výpočet funkční hodnoty řadou, iterací, vzorce a pod.)

6. Příkaz DO. Dovolené způsoby zápisu a vzájemného umístění cyklů. Standardní funkce FORTRANu. Pojem základní a generované funkce. Základní informace o příkazech READ a WRITE.

7. Opakování příkazu DO. Pokračování o příkazech READ a WRITE. Jednoduchý seznam a jeho prvky. Cyklický seznam. Příkaz FORMAT. Konverze I, F, H. Pojem záznamu. Řízení tiskárny prvními znaky záznamu.

8. Opakování příkazů READ, WRITE a FORMAT. Konverze X, apostrofová konverze ^M). Nadočerpání a opakované čerpání formátové specifikace bez závorek. Oddělovač /. Volná vstupní konverze IG, FG, G ^M). Překlad a sestavení programu konkrétně, řídicí příkazy pro překlad, sestavení a výpočet.

9. Další podrobnosti o příkazech READ a WRITE. Závorky ve formátové specifikaci. Předvedení kompletního příkladu programu a skutečného protokolu o překladu a sestavení programu. Ukázka skutečných výsledků výpočtu získaných tímto programem.