TECHNICKÁ A SYSTÉMOVÁ PODPORA VÝUKY OPERAČNÍCH SYSTÉMŮ (NA FIT ČVUT v Praze)

Ing. Zdeněk Muzikář, CSc.
Katedra počítačových systémů
Fakulta informačních technologií ČVUT
Thákurova 9, 160 00 Praha 6
e-mail: muzikar@fit.cvut.cz

Doc. RNDr. Ing. Petr Zemánek, CSc.
Katedra počítačových systémů
Fakulta informačních technologií ČVUT
Thákurova 9, 160 00 Praha 6
e-mail: petr.zemanek@cvut.cz

Princip výuky operačních systémů

- Sekvence navazujících předmětů
- Vzrůstající obtížnost
- Teoreticky orientované
- Prakticky orientované

- Ověření znalostí
  - Teoretické předměty – zkouška (ústní, písemný klasický nebo e-test)
  - Praktické předměty – nutnost **ověření správnosti praktické úlohy**,
    tj. funkčnosti příkazu, programu/skriptu, konfigurace systému, sítě, disku apod.

Ověřování správnosti příkazu nebo programu

Zkušenost:
Příklady by měly být realistické (přiměřeně složité) a hodnocení by mělo být škálované

„Ruční“ kontrola

- Časově náročné, subjektivní, náchylné k omylům **–**
- Hodnocení lze škálovat (dle kvality programu) **+**
- Při velkém množství studentů a/nebo složitějších programech prakticky nepoužitelné **–**

Automatická kontrola (správnost výstupních dat)

- Rychlé a objektivní **+**
- Výsledek ano/ne – neumožňuje škálování **–**

Závěr:
Automatická kontrola umožňující škálování → speciální systém

Ověřování správnosti praktických úloh

Problémy:

1. **Ověření správnosti příkazu nebo programu**
   - „Ruční“ kontrola
   - Automaticky (kontrola správnosti výstupních dat)

2. **Možnost prakticky testovat potenciálně destruktivní aktivity**
   - Zničení systému
   - Nutnost (rychlé!) obnovy

Systém LearnShell

Princip:

- Generování sekvence **variant vstupních dat**
- Varianty mají narůstající obtížnost
- Varianty jsou postupně zpracovávány testovaným a referenčním skriptem
- Výstupní data se porovnají a při shodě navyšují skóre testovaného skriptu

Učitel připraví referenční řešení, popisy variant vstupních dat a ohodnocení každé varianty.

Varianty se generují v několika **instancích** podle popisu, každá instance obsahuje **náhodné** hodnoty – viz příklad.

Celkové hodnocení je součtem hodnocení úspěšně vyřešených variant → škálování

Systém LearnShell – jednoduchý příklad

Zadání úlohy:

Máme soubor obsahující IPV4 adresy, vždy jedna adresa na řádku. Napište skript, který vypíše všechny adresy, jejichž první byte je shodný s parametrem skriptu. Maximální získaný počet bodů je 5.

Vyhodnocování:

Skript je testován pomocí **parametru,** který je náhodně generován z předdefinovaného rozsahu. Vstupní soubor je generován také náhodně, ale v návaznosti na vygenerovaný parametr, podle popsaných pravidel pro každou variantu vstupních dat.
Úspěšné řešení varianty (stejný výstup jako referenční řešení) je ohodnoceno určitým počtem bodů a poté se pokračuje další variantou. Při neúspěšném řešení testování končí a hodnocení je dáno součtem dosud získaných bodů.

Systém LearnShell – příklad, varianta 1

Pozn.: **Hodnota parametru** pro všechny následující případy bude **19,** což je zjednodušení, neboť se generuje vždy náhodně. Další zjednodušení je, že varianta nemá více náhodných instancí (pro splnění musí být vyřešeny všechny).

Vstupní soubor pro tuto variantu:

```
10.11.12.13        # neodpovídá parametru
14.15.16.17        # neodpovídá parametru
19.20.1.1          # odpovídá parametru
```

Jako správná řešení jsou v tomto případě uznána i naivní řešení typu **grep $1** nebo **grep $1.** a ohodnocena např. 0.5 bodem. Řešení je sice obecně nesprávné, ohodnocení minimální, ale motivační a odliší „alespoň něco“ od totálního nesmyslu

Vyhodnocování pak pokračuje další variantou.

Systém LearnShell – příklad, varianta 2

Vstupní soubor pro tuto variantu:

```
10.11.12.13          # neodpovídá parametru
14.15.16.17          # neodpovídá parametru
19.20.1.1            # odpovídá parametru
20.19.1.1            # neodpovídá parametru
```

Předchozí řešení je v tomto špatně a jako správné řešení bude uznáno např. **grep “^$1“** (i když obecně stále správné není) a bude ohodnoceno např. 1.5 bodem.

Celkově získaný počet bodů bude 2 a vyhodnocování pokračuje další variantou.

Systém LearnShell – příklad, varianta 3

Vstupní soubor pro tuto variantu:

```
10.11.12.13       # neodpovídá parametru
14.15.16.17       # neodpovídá parametru
19.20.1.1         # odpovídá parametru
20.19.1.1         # neodpovídá parametru
192.168.1.1       # neodpovídá parametru
```

Všechna předchozí řešení jsou nyní špatně, jediné správné řešení je **grep “^$1\.“** a bude ohodnoceno 3 body.

Celkový a maximální počet bodů za správné řešení bude 5.

Možnost prakticky testovat destruktivní aktivity

Zkušenost:
Výuka administrace OS bez praktického zkoušení nemá smysl.

- Nutnost práce v administrátorském režimu
- Změna konfigurace systému a jeho komponent (disky, síťová rozhraní,…)
- Vysoká pravděpodobnost destrukce systému

Možná řešení:

- Student trvale používá „svůj“ počítač, který se po ukončení kurzu standardně přeinstaluje (typické při několikadenních školeních v komerčních firmách).
- V univerzitním prostředí (krátké výukové bloky 1x týdně po dobu semestru) vede k nutnosti rychlé přeinstalace mnoha počítačů během max. 10 min (přestávka).
    - Virtualizace (neřeší vše)
    - Systém rychlé přeinstalace → tzv. „bourací učebna“ (řešení FIT)

Systém rychlé přeinstalace I

- Počítače učebny jsou připojeny ke vzdálenému boot serveru
- Boot server je zároveň úložištěm připravených obrazů systémů
- Síťový boot zavede operační systém a spustí se obslužný program
- Nezáleží na stavu počítače, podmínkou je pouze funkčnost síťové karty
- Obslužný program pracuje v režimu UPLOAD (uloží obraz připraveného systému) a DOWNLOAD (stáhne a spustí obraz vybraného systému)

Vlastnosti

- Rychlá přeinstalace (5-10 min)
- HW konfigurace počítačů musí být identická
- Nutný firewall (na všech systémech pracují neidentifikovatelní admini)

Systém rychlé přeinstalace II

UPLOAD (Vytvoření a uložení obrazu systému)

- Uživatel (učitel) nainstaluje a nakonfiguruje na počítači libovolný systém
  - Např. image zaměrně poškozeného systému – vhodné pro troubleshooting
- Provede síťový boot
- V obslužném programu zvolí možnost UPLOAD
- Definuje identifikaci (jméno) obrazu

Princip:

Obslužný program fyzicky přečte systémový disk včetně partition tabulky, zkomprimuje a vše uloží jako soubor (obraz systému) na boot serveru pod zadaným jménem.

Systém rychlé přeinstalace III

DOWNLOAD (stažení obrazu systému a spuštění)

- Uživatel (student) provede síťový boot a v obslužném programu zvolí možnost DOWNLOAD
- Ze seznamu vybere podle jména požadovaný soubor (systém)

Princip:

Obslužný program po síti stáhne se serveru vybraný soubor a fyzicky jím přepíše systémový disk včetně partition tabulky. Na závěr provede operaci reboot.

Závěr

Oba popsané nástroje se osvědčily a umožňují kvalitní praktickou výuku operačních systémů ve specifickém prostředí FIT ČVUT.

LearnShell umožňuje snadné, frekventované, a přitom objektivní a motivující hodnocení velkého množství studentů.

Specializovaná **„bourací“ učebna** umožňuje plnohodnotné zkoušení administrátorských aktivit. Navíc se začala přirozeně využívat i v jiných předmětech, kde je nutná specifická konfigurace systému nebo instalace speciálního software.