Přednáška 2

Zpracování příkazové řádky. Speciální znaky.

Katedra počítačových systémů FIT, České vysoké učení technické v Praze

Příprava studijního programu Informatika je podporována projektem financovaným z Evropského sociálního fondu a rozpočtu hlavního města Prahy.
Praha & EU: Investujeme do vaší budoucnosti

Příkazová řádka – motivace

Přiřazení hodnoty do proměnné

```
A=Dobry den
A="Pocet prihlasenych  uzivatelu:    `finger | tail +2 | wc –l` "
```

Volání příkazu/příkazů

```
ls   –la   /
echo   $A
finger   |   tail +2   |   wc –l

:(){ [ $1 -gt $2 ] && echo $1 || echo $2; }; : 1 3; : 2 1
```

Zpracování příkazové řádky

1. Detekce znaků rušících speciální význam znaků

\	ruší speciální význam následujícího znaku
´ … ´	všechny znaky uzavřené mezi apostrofy ztrácejí speciální význam (kromě apostrofu)
" … "	mezi uvozovkami ztrácí speciální význam všechny znaky kromě: \ ruší význam následujícího znaku ` příkaz ` náhrada příkazů $ náhrada obsahu proměnné aritmetický výraz (mimo sh)

– Shell tyto znaky odstraní při interpretaci řádky.

Zpracování příkazové řádky

2. **Odstranění komentářů** (#)
3. **Postupné rozdělení příkazové řádky na jednoduché příkazy a přesměrování vstupu/výstupu**

- **jednoduchý příkaz (simple command)**
  - posloupnost přiřazení hodnot proměnným oddělených mezerami nebo tabelátory
  - jméno příkazu následované jednotlivými argumenty
- **roura (pipeline)**
  - posloupnost jednoho nebo více příkazů oddělených operátorem |
- **seznam příkazů (list)**
  - posloupnost jedné nebo více rour oddělených operátory ; & && || a ukončených operátory ; & <newline>
- **složený příkaz (compound command)**

  (list) { list ; } ((výraz)) [[výraz]] for while until if case

Zpracování příkazové řádky

- Každý příkaz vrací **návratový kód**

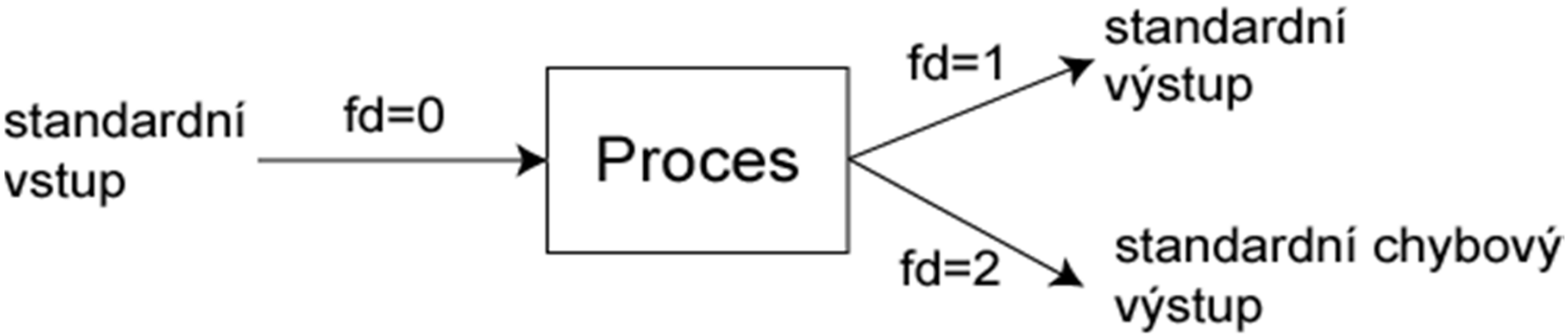
  (0 = úspěšné provedení, 1,…,255 = chyba)

příkaz &	asynchronní provádění příkazu shell nečeká na jeho dokončení, příkaz se provádí „na pozadí“)
příkaz1 ; příkaz2	sekvenční provádění příkazů, nejdříve se provedou příkazy před a pak příkazy za středníkem
(příkaz1 ; příkaz2)	podshell, příkazy jsou spuštěny v nové instanci shellu
{ příkaz1 ; příkaz2 }	příkazy jsou spuštěny v aktuální instanci shellu

Zpracování příkazové řádky

příkaz1 | příkaz2	roura, příkazy se startují zleva a běží paralelně, standardní výstup předchozího je standarním vstupem následujícího příkazu, návratový kód roury je návratovým kódem posledního příkazu
příkaz1 && příkaz2	sekvenční provádění příkazů, příkaz „za“ se provede pouze tehdy, vrátí-li příkaz „před“ nulový návratový kód (skončí bez chyby)
příkaz1 || příkaz2	sekvenční provádění příkazů, příkaz „za“ se provede pouze tehdy, vrátí-li příkaz „před“ nenulový návratový kód (skončí s chybou)

Přesměrování vstupu/výstupu



- Procesy přistupují k souborům pomocí tzv. deskriptorů souborů (0,1,2,…).
- Každý proces má při spuštění standardně otevřeny tyto deskriptory:
  0 – standardní vstup
  1 – standardní výstup
  2 – standardní chybový výstup
- Nový proces standardně dědí deskriptory souborů od svého rodiče.
- Pomocí speciálních znaků lze v shellu předefinovat jednotlivé deskriptory.

Zpracování příkazové řádky

příkaz < soubor	soubor bude otevřen a nastaven jako std. vstup příkazu
příkaz > soubor	soubor bude otevřen jako std. výstup z příkazu pokud soubor neexistuje, bude otevřen pokud existuje, bude přepsán (lze potlačit nastavením parametru noclobber v shellu – mimo sh)
příkaz >> soubor	soubor bude otevřen jako std. výstup z příkazu pokud soubor neexistuje, bude otevřen pokud existuje, bude výstup připojen na konec
příkaz << řetězec	shell čte vstup až do řádky začínající daným řetězcem (tzv. „here-document“) načtený text se stane std. vstupem příkazu

Zpracování příkazové řádky

příkaz 2> soubor	soubor bude otevřen jako std. chybový výstup z příkazu pokud soubor neexistuje, bude otevřen pokud existuje, bude přepsán
příkaz >&*n*	std. výstup bude zapsán do souboru určeného deskriptorem *n*
příkaz *m*>&*n*	deskriptor *n* se přiřadí do deskriptoru *m* výstup příkazu do souboru určeného deskriptorem *m* se přesměruje do souboru určeného deskriptorem *n*
příkaz > soubor 2>&1	Std. výstup i std. chybový výstup bude zapsán do souboru.

- Při vícenásobném přesměrování se přesměrování vyhodnocují zleva doprava.

Zpracování příkazové řádky

4. Náhrada

- aliasů (zkratek příkazů)
- znaku tilda (~)

~	odpovídá domovskému adresáři (kromě sh)
~uživatel	odpovídá domovskému adresáři daného uživatele (kromě sh)

- příkazů

` příkaz `	příkaz mezi opačnými apostrofy je proveden a nahrazen (včetně těchto apostrofů) svým std. výstupem
$(příkaz)	to samé, pouze jiná syntaxe (mimo sh)

- aritmetických výrazů $((výraz))
- proměnných ($1, $HOME,…)

5. Rozdělení na slova

newline mezera TAB	oddělovače slov (lze změnit pomocí proměnné IFS)

6. Náhrada jmen souborů

*	odpovídá libovolnému řetězci kromě . na začátku a / kdekoliv
?	odpovídá jednomu libovolnému znaku kromě . na začátku a / kdekoliv
[abc] [a-z]	odpovídá jednomu znaku z uvedených znaků resp. z uvedeného intervalu
[!abc] [!a-z] [^ijk] [^m-z]	odpovídá jednomu znaku mimo uvedených znaků (varianta s ^ není v sh)

- Znaky . na začátku jména a / se musí explicitně uvádět.
- Pokud ničemu neodpovídá, pak text zůstává nezměněn.
- Nahrazování *, ? a [ ] lze potlačit příkazem `set -f` a opět povolit příkazem `set +f` (nedoporučuje se).

Zpracování příkazové řádky

7. **Nastavení parametrů**
8. **Přiřazení hodnot proměnným nebo volání příkazů**

Příkaz eval řetězec

- aplikuje kroky 1-8 na zadaný řetězec

Příklad:

```
unset B  ;  A='$B'  ;  B=xzy  ;  echo $A

unset B  ;  A='$B'  ;  B=xzy  ;  eval echo $A

echo "Zadej prikaz: "  ;   read A  ;  eval $A
```

Volání příkazu

- **Hledání příkazu**
  - absolutní/relativní cesta k příkazu
  - funkce
  - vnitřní příkaz interpretu
  - pokud je příkaz zadán pouze jménem a není to funkce ani vnitřní příkaz, pak se hledá první výskyt spustitelného programu v adresářích definovaných v proměnné PATH zleva doprava

Proměnné shellu

proměnná=hodnota	přiřadí hodnotu do proměnné
$proměnná	dosadí se obsah proměnné

- **Příklady vnitřních proměnných shellu**

PS1	definice promptu
PAGER	příkaz zobrazující manuálové stránky
PATH	seznam adresářů, kde se hledají příkazy
MANPATH	seznam adresářů, kde se hledají man. stránky

Proměnné shellu

- **Příklady vnitřních proměnných shellu**

HOME	domovský adresář
PWD	aktuální adresář

0	jméno příkazu
1,…,9	poziční parametr
#	počet pozičních parametrů na příkazové řádce
$	identifikační číslo procesu (PID) dané instance shellu
?	návratový kód právě ukončeného procesu