1. projekt 2010/2011

Podmínky vypracování

- Z interpretů můžete použít .*sh, awk (gawk) i sed. Dále standardní utility pro práci s textem (grep, cat, cut, sort, uniq, apod.).
- Nepoužívejte tzv. bashismy (rozšíření typické pro bash) - vyzkoušejte si vaše konstrukce v ksh na stroji eva.fit.vutbr.cz (FreeBSD), případně krok.fit.vutbr.cz (OpenSolaris).
- Dodržujte syntax zadaných jmen, formátu souborů a formátu výstupních dat! Čtěte pozorně zadání a poznámky k vypracování u jednotlivých zadání.
- **Dotazy k zadání:** Veškeré nejasnosti a dotazy řešte pouze prostřednictvím diskuzního fóra *Projekty*.

Odevzdání

- Vytvořený skript zabalte do archivu zip, jméno archivu bude xlogin.zip, kde xlogin je váš login.
- Skripty jsou Unixové textové soubory (řádky jsou ukončeny znakem 0x0a), při vypracování úloh na Windows doporučujeme před odevzdáním použít utilitu dos2unix.
- Archiv xlogin.zip odevzdejte prostřednictvím informačního systému, termín *Projekt 1*.
- Pokud nebude dodržena forma odevzdání nebo nepůjde skript spustit, může být projekt hodnocen 0 body.
- Komentáře nebo hlášení pište anglicky nebo česky/slovensky bez diakritiky.

Cílem úkolu je vytvořit 2 skripty. První skript pokrývá práci se soubory archivů a druhý skript se zabývá textovými filtry (zvýraznění syntaxe).

A. Práce s archivy

Cíl: Mějme dvě verze jednoho software. Zdrojové kódy každé verze jsou zabaleny v jednom archivu. Vašim cílem je vytvořit skript *ardiff*, který (1) zaznamenává změny souborů dvou různých archivů, (2) tyto změny ukládá do adresářové struktury odpovídající daným archivům a (3) aplikuje změny na jednom z těchto archivů pro získání archivu druhého.

Popis funkčnosti:

1. **Zaznamenání změn:** Skript *ardiff* dostane jména dvou různých archivů a vypíše seznam souborů, které se v archivech liší (tj. liší se jejich obsah nebo v existují pouze v jednom archivu). Každý soubor bude vypsán na jeden řádek včetně cesty v archivu.
2. **Uložení změn:** Skript dostane jména dvou různých archivů a vytvoří tzv. *rozdílový archiv*. Adresářová struktura rozdílového archivu bude odpovídat oběma zdrojovým archivům (tj. adresář vyskytující se v rozdílovém archivu musí mít odpovídající adresář alespoň v jednom zdrojovém archivu). Každý regulární soubor cílového rozdílového archivu bude obsahovat pouze rozdíly daných souborů archivů zdrojových a bude mít příponu .patch. Soubory, které jsou v obou zdrojových archivech shodné, nebudou mít v rozdílovém archivu žádný rozdílový soubor. Rozdílové soubory musí odpovídat sjednocenému kontextu s implicitním nastavením (angl. unified context) viz *diff(1)*. Názvy porovnávaných souborů (v hlavičce každého rozdílového souboru) budou včetně cesty v daném zdrojovém archivu a cesta bude začínat "a/" v případě prvního zdrojového archivu a "b/" v případě druhého zdrojového archivu (viz příklad diff.zip). Skript se musí vyrovnat i se soubory, které existují pouze v jednom archivu (s neexistujícími soubory v druhém archivu nakládejte jako s prázdnými soubory) nebo se souborem, který je v jednom zdrojovém archivu regulární a ve druhém adresář. Výsledný archiv nebude obsahovat prázdné soubory a adresáře.
3. **Aplikace změn:** Skript obdrží jméno jednoho zdrojového archivu a rozdílového archivu a pomocí aplikace změn uložených v rozdílovém archivu nad zadaným zdrojovým archivem vytvoří archiv odpovídající druhému archivu. Implicitně se předpokládá dopředná aplikace změn (skript obdrží starou verzi a rozdílový archiv a cílem je vytvořit archiv nové verze). Parametrem -r se skript přepíná do reverzního režimu (z nové verze a rozdílového archivu vytvoří archiv odpovídající staré verzi). Výsledný archiv nebude obsahovat prázdné

soubory a adresáře.

4. Skript musí podporovat archivy typu **tar** (nekomprimovaný nebo komprimovaný pomocí gzip nebo bzip2) a **zip**. Typ existujícího archivu skript pozná podle obsahu, nikoliv podle přípony. Typ cílového archivu (rozdílový archiv v případě uložení změn nebo druhý archiv v případě aplikace změn) skript pozná podle přípony zadaného jména archivu: .tar -> POSIX tar, .tgz nebo .tar.gz -> gzip zabaleny POSIX tar, .tar.bz2 -> bzip2 zabalený POSIX tar, .zip nebo .ZIP -> Zip archiv.
5. Neočekávané chování řešte způsobem obvyklým pro unixové nástroje: při neočekávaném ukončení (konkrétně uživatelem klávesou Ctrl-C, systémem např. při shutdown a nebo odpojení terminálu) musí skript odstranit všechny dočasně vytvořené soubory a ukončit se s chybovým návratovým kódem a při jakékoliv chybě v průběhu skriptu musí ještě vypsat hlášení na chybový výstup.
6. Skript během své bezproblémové činnosti nic nevypisuje. Všechny dočasné soubory budou odstraněny.

Použití skriptu:

```
$ ./ardiff
ardiff vypisuje zmeny archivu, vytvari rozdilovy archiv nebo aplikuje rozdilovy
archiv na zdrojovy archiv.
Pouziti: ardiff [volby] archiv1 archiv2
Volby:
  -o SOUBOR  Pokud je cilem skriptu vytvorit archiv, bude vytvoren do souboru
             se jmenem SOUBOR (plati pro -c a -p).
  -l         Vypis seznamu souboru, ktere se v zadanych archivech lisi.
  -c         Vytvoreni rozdiloveho archivu.
  -p         Aplikace rozdiloveho archivu (argument archiv2) na zdrojovy archiv
             (argument archiv1).
  -r         Prepnuti do reverzniho rezimu (plati pro -p).
```

Poznámky k vypracování:

- Se všemi soubory v archivech pracujte jako s textovými soubory.
- Se symbolickými odkazy pracujte jako se soubory, na které odkazují.
- Předpokládejte, že zdrojové archivy neobsahují prázdné soubory a adresáře.
- Dočasné soubory můžete vytvářet pouze ve vaším skriptem vytvořeném adresáři, který může být umístěn pouze v adresáři /tmp.
- Pro zpracování parametrů vašeho skriptu použijte *getopt(1)* nebo *getopts(1)*. Při chybějících nebo špatně zadaných parametrech skript vypíše hlášení/nápovědu o svém použití, viz text výše (Použití skriptu).
- Skript nesmí být za žádných okolností interaktivní.

Příklad:

```
$ ./ardiff -l old.tgz new.tar.bz2
hello/LICENSE
hello/doc/README
hello/doc/README/make
hello/doc/README/fig2dev
hello/doc/index.html
hello/doc/schello.fig
hello/makefile
hello/hello.c
hello/dir/changed/current
ver
$ ./ardiff -c -o diff.zip old.tgz new.tar.bz2
$ ./ardiff -p -r -o old.zip new.tar.bz2 diff.zip
```

Zde jsou odpovídající archivy: old.tgz, new.tar.bz2, diff.zip a old.zip (obsah old.tgz a old.zip mají být po zpětné aplikaci rozdílového archivu nad new.tar.bz2 stejné).

B. Zvýraznění syntaxe

Cíl: Skript *hltrace* zvýrazňuje klíčové prvky stopy (angl. trace) zaznamenané pomocí nástroje *strace(1)*. Skript

pracuje jako textový filtr. Vstupem skriptu je stopa z programu strace, výstupem je obsah HTML stránky obsahující stopu se zvýrazněnou syntaxí.

Popis funkčnosti:

- Na začátek výsledného HTML skript vloží následující řádky:

```
<html>
<style>
.pid { color:darkred; }
.ts { color:navy; }
.number { color:red; }
.const { color:green; }
.string { color:blue; }
.hlcall { text-decoration:none; font-weight:bold; color:black; }
.call { text-decoration:none; color:olive; }

</style>
<body><pre>
```

- Následuje stopa se zvýrazněnou syntaxí:
  - Všechny výskyty znaků '<', '>' a '&' nahraďte HTML znaky "<", ">" a "&".
  - Každý zvýrazněný prvek (kromě názvu systémového volání) bude vložen mezi značky:

    ```
    <span class="CLS"> ... </span>
    ```

    kde CLS označuje třídu zvýraznění.

  - Třídy zvýraznění jsou následující:
    - Pokud stopa obsahuje číslo procesů (strace s parametrem -f), tato čísla budou zvýrazněna třídou **pid**.
    - Pokud stopa obsahuje relativní časové údaje (strace s parametrem -r), každá časová snačka bude zvýrazněna třídou **ts**.
    - Číselné parametry nebo návratové kódy (v desítkové nebo hexadecimální soustavě) systémových volání budou označena třídou **number**. Uvažujte pouze samostatné parametry, nikoliv číselné hodnoty parametru strukturovaného typu (viz např. volání *fstat(2)*, tj. viz strace -e trace=fstat true).
    - Pojmenované konstanty (NULL, O_RDONLY apod., tj. identifikátory začínající velkým písmenem a skládající se z velkých písmen, číslic a podtržítek) patří do třídy **const**.
    - Řetězcové literály (odpovídají jednořádkovým řetězcovým literálům jazyka C) patří do třídy **string**.
  - Každé systémové volání bude odkaz na patřičnou manuálovou stránku, tedy jeho název bude vložen mezi značky:

    ```
    <a href="http://www.kernel.org/doc/man-pages/online/pages/man2/SYSCALL.2.html" class="CLS"> ... </a>
    ```

    kde SYSCALL je jméno systémového volání a CLS je třída zvýraznění syntaxe:

    - Uživatelem definované systémové volání bude zvýrazněno třídou **hlcall** (viz níže Použití skriptu).
    - Všechna ostatní volání budou zvýrazněna třídou **call**.
- Na konec HTML skript vloží jeden řádek

```
</pre></body></html>
```

Použití skriptu: Pozor změna! (9.3.2011)

```
$ ./hltrace -h
hltrace zvyrazni syntax stopy od strace.
Pouziti: hltrace [volby] <stopa.strace >stopa.html
Volby:
  -s SYSCALL  Specialne zvyrazni volani SYSCALL.
```

Poznámky k vypracování:

- Je zakázáno používat dočasné soubory.
- Očekávejte stopu kompatibilní se stopou získanou programem strace s parametrem -o (příp. v kombinaci s -f nebo -r).

Příklad:

```
$ strace -r -f -o sh.txt -e trace=process sh -c "cat /etc/fstab|grep ext" >/dev/null
$ ./hltrace -s execve <sh.trace >sh.html
```

Zde jsou odpovídající soubory: sh.trace a sh.html.

nebo:

```
strace -e trace=ipc ipcs 2>&1 >/dev/null | ./hltrace >ipcs.html
```