

vyuka:ios_ulohy

1. Úloha (2013)

Poznámka: **Takto je označen text, který se v zadání v průběhu vypracování úlohy změnil.**

Odevzdání

- Součástí odevzdání budou pouze skripty `le-checkout.sh`, `le-update.sh`, `le-revert.sh` a `le-diff.sh`. Tyto soubory zabalte pomocí nástroje `zip` do archivu s názvem `xlogin.zip`, kde `xlogin` je váš login.
- Archiv vytvořte tak, aby po rozbalení `unzip xlogin.zip` byly soubory `le-*.sh` umístěny ve stejném adresáři, jako je archiv.
- Archiv `xlogin.zip` odevzdejte prostřednictvím informačního systému, termín Projekt 1.
- Pokud nebude dodržena forma odevzdání, může být úloha hodnocena 0 body.

Popis úlohy

Cílem úlohy je vytvořit sadu skriptů pro podporu lokálních modifikací zdrojových kódů získaných ze sdíleného úložiště. Tyto skripty mohou sloužit např. pro podporu experimentů s takovými zdrojovými kódy. Předpokládá se práce nad třemi adresáři: tzv. *adresář projektu* (adresář sdílený více uživateli), *experimentální kopie* (soubory určené pro experimentální úpravy) a *referenční kopie* (soubory sloužící pro identifikaci stavu, ze kterého začal uživatel provádět experimenty). Sada skriptů umožňuje udržovat synchronizovaný stav mezi těmito adresáři, aplikovat změny provedené v adresáři projektu na experimentální kopii a zobrazovat změny provedené v experimentální kopii. Sada skriptů bude obsahovat skripty: `le-checkout.sh`, `le-update.sh`, `le-revert.sh` a `le-diff.sh` (předpona `le-` je zkratka pro Local Experiments).

Specifikace chování skriptů

- Skripty (kromě `le-checkout.sh`) zpracovávají soubory, jejichž jména byla zadána jako argumenty skriptu. V případě, že není zadáno žádné jméno souboru, skripty zpracovávají celý obsah adresáře projektu nebo referenční/experimentální kopie.
- Všechny skripty budou spouštěny z adresáře experimentální kopie.
- Adresář referenční kopie má jméno `.le` a je umístěn v adresáři experimentální kopie.
- Skripty se řídí obsahem konfiguračního souboru `.config` uloženého v adresáři referenční kopie. Konfigurační soubor musí obsahovat umístění adresáře projektu a může obsahovat seznam regulárních výrazů vyjadřující jména souborů, která se při jakýchkoliv operacích mají ignorovat. Formát konfiguračního souboru je popsán níže.
- Skripty pracují nad adresáři s hloubkou 1. Skripty tedy ignorují podadresáře adresáře projektu a podadresáře referenční a experimentální kopie. Skripty navíc ignorují skryté soubory (kromě konfiguračního souboru).
- Skripty na `stdout` vypisují pouze hlášení zmíněná v zadání.

- Skripty mohou používat dočasné soubory, ale po ukončení skriptů (i při neúspěchu) musí být dočasné soubory smazány. K vytváření dočasných souborů je doporučeno použít aktuální adresář nebo /tmp.

le-checkout.sh

Předpokládaný rozsah: 20-40 řádků

Spuštění: `le-checkout.sh projdir`

Popis: Argument `projdir` je cesta k adresáři projektu (reprezentujícímu sdílené úložiště). Aktuální adresář, v kterém je skript spuštěn, se považuje za adresář experimentální kopie. Skript v tomto adresáři vytvoří podadresář `.le` pro referenční kopii (pokud tento adresář již existuje, **smaže v něm všechny soubory, které nejsou skryté bude nejprve smazán**). Ostatní soubory a podadresáře, které se v adresáři experimentální kopie již nachází, skript ponechá beze změn. V dalším kroku skript zkopíruje všechny soubory z adresáře projektu do referenční a experimentální kopie. Do konfiguračního souboru následně uloží informaci `projdir` (viz popis konfiguračního souboru `.config`). Pokud konfigurační soubor existoval už před spuštěním `le-checkout.sh` a informaci o adresáři projektu již obsahoval, bude tato informace přepsána na hodnotu zadanou argumentem `projdir`. Navíc, pokud konfigurační soubor již existoval, bude celý skript respektovat informaci `ignore` v něm uloženou.

le-update.sh

Předpokládaný rozsah: 60-100 řádků

Spuštění: `le-update.sh [FILE] ...`

Popis: Skript provede aktualizaci referenční a experimentální kopie, do které zahrne poslední úpravy provedené v adresáři projektu (např. jiným uživatelem) a prozatímní úpravy v experimentální kopii. Pro každý soubor v adresáři projektu, referenční nebo experimentální kopii mohou při aktualizaci nastat následující případy:

1. Soubor je stejný v adresáři projektu a v referenční a experimentální kopii: Skript vypíše `. : jméno souboru (tečka dvojtečka mezera jméno_souboru znak_konce_řádku)`.
2. Soubor je stejný v adresáři projektu a v referenční kopii, ale v experimentální kopii se liší: Skript vypíše `M: jméno souboru`.
3. Soubor je stejný v adresáři projektu a v experimentální kopii, ale liší se v referenční kopii: Skript vypíše `UM: jméno souboru` a zkopíruje soubor z adresáře projektu do referenční kopie.
4. Soubor je stejný v referenční a experimentální kopii, ale liší se od souboru v adresáři projektu: Skript zkopíruje soubor z adresáře projektu do experimentální a referenční kopie a vypíše `U: jméno souboru`.
5. Soubor se liší ve všech umístěních - skript provede sloučení změn:
 - a. Nejprve zjistí, jaké změny byly provedeny v adresáři projektu oproti referenční kopii (nápopověda: použijte `diff(1)` [<http://pubs.opengroup.org/onlinepubs/009695399/utilities/diff.html>] s parametrem `-u`),
 - b. potom se změny pokusí aplikovat do experimentální kopie (nápopověda: použijte `patch(1)` [<http://pubs.opengroup.org/onlinepubs/009695399/utilities/patch.html>]). Na základě výsledku mohou nastat následující případy:
 - I. Aplikace změn proběhla úspěšně: Skript uloží výsledek sloučení změn do souboru v

experimentální kopii, výpíše M+: jméno souboru a zkopíruje soubor z adresáře projektu do referenční kopie.

II. Aplikace změn se nepovedla: Skript ponechá soubor v experimentální a referenční kopii bez změn a vypíše M!: jméno souboru `conflict!` (Pozn.: V experimentální ani v referenční kopii nesmí zůstat žádné pomocné soubory).

6. Soubor existuje v adresáři projektu, ale chybí v experimentální kopii: Skript zkopíruje soubor z adresáře projektu do experimentální a referenční kopie a vypíše C: jméno souboru.
7. Soubor existuje v referenční kopii, ale již neexistuje v adresáři projektu: Skript vypíše D: jméno souboru a smaže soubor v experimentální a referenční kopii.
8. Všechny ostatní případy jsou skriptem ignorovány.

le-revert.sh

Předpokládaný rozsah: 15-25 řádků

Spuštění: `le-revert.sh [FILE] ...`

Popis: Skript kopíruje obsah referenční kopie do experimentální kopie. Všechny soubory, které jsou v experimentální kopii navíc, budou ponechány.

le-diff.sh

Předpokládaný rozsah: 20-40 řádků

Spuštění: `le-diff.sh [FILE] ...`

Popis: Skript na standardní výstup vypíše změny mezi adresářem projektu a experimentální kopii. Pro výpis změn použijte `diff(1)` [<http://pubs.opengroup.org/onlinepubs/009695399/utilities/diff.html>] s parametry -u proj exp, kde proj je soubor v adresáři projektu a exp je soubor v experimentální kopii. Pokud se soubory neliší, skript vypíše .: jméno souboru. Pokud soubor existuje pouze v adresáři projektu, místo změn skript vypíše C: jméno souboru. Pokud soubor existuje pouze v experimentální kopii, místo změn skript vypíše D: jméno souboru.

Konfigurační soubor .config

Konfigurační soubor `.config` se nachází v adresáři referenční kopie. Jedná se o textový soubor obsahující konfigurační řádky. Konfigurační řádek obsahuje umístění adresáře projektu nebo regulární výraz vyjadřující soubory, které mají být všemi skripty ignorovány. Regulárních výrazů může být více. Pořadí jednotlivých konfiguračních řádků není definováno.

1. Umístění adresáře projektu: řádek ve tvaru `projdir path/to/dir`, kde `path/to/dir` je relativní nebo absolutní cesta k adresáři projektu.
2. Seznam regulárních výrazů: řádky ve formátu `ignore EREGEXP`, kde `EREGEXP` je rozšířený regulární výraz (pro `grep(1)` [<http://pubs.opengroup.org/onlinepubs/009695399/utilities/grep.html>] s argumentem -E).

Příklad konfiguračního souboru:

```
ignore \..*
projdir ../clonedproject
ignore README
ignore License
ignore .*\.o
```

Příklady použití

Pozn.: Před zkoušením vašeho řešení doporučujeme do proměnné PATH přidat cestu k vyvíjeným skriptům:

```
$ export PATH=/path/to/IOS/skripts:$PATH
```

1. mějme adresář projektu:

```
$ ls /path/to/proj
f1.txt
$ cat /path/to/proj/f1.txt
prvni
druhy
treti
ctvrty
paty
```

2. vytvoření experimentální kopie:

```
$ cd /path/to/exp
$ le-checkout.sh /path/to/proj
$ ls -A
.le    f1.txt
$ cat .le/.config
projdir /path/to/proj
$ ls -A .le
.config  f1.txt
```

3. provedení nějakých úprav v experimentální kopii (dopíšeme slovo radek na první řádek):

```
$ vim f1.txt
$ cat f1.txt
prvni radek
druhy
treti
ctvrty
paty
```

4. mezitím někdo provedl úpravu v adresáři projektu:

```
$ cat /path/to/proj/f1.txt
prvni
druhy
treti
ctvrty
paty
sesty radek
```

5. provedeme spojení změn:

```
$ le-update.sh
M+: f1.txt
$ cat f1.txt
prvni radek
druhy
treti
ctvrty
paty
sesty radek
```

6. v čem se nyní liší náš experiment a upravený adresář projektu:

```
$ le-diff.sh
--- /path/to/proj/f1.txt      2013-02-18 15:27:33.999355004 +0100
+++ f1.txt      2013-02-18 15:27:37.729373912 +0100
@@ -1,4 +1,4 @@
-prvni
+prvni radek
  druhy
   trei
   ctvrty
```

7. návrat k verzi posledního update:

```
$ le-revert.sh
```

8. nyní diff ukáže, že f1.txt je stejný jako v adresáři projektu:

```
$ le-diff.sh
.: f1.txt
$ cat f1.txt
prvni
druhy
treti
ctvrty
paty
sesty radek
```

Základní testovací sada

Hodnoceny mohou být i částečně vypracované domácí úkoly. Hodnocení bude záviset na náročnosti vypracovaných částí. Příklad možných částečných vypracování úlohy:

1. odevzdání pouze vybraných skriptů
2. podpora pouze jednoho konfiguračního řádku ignore
3. žádná podpora konfiguračního řádku ignore
4. žádná podpora výběru souborů zadaných argumenty skriptů

Stáhněte si jednoduchý testovací skript [test.sh](#) do adresáře s vašimi skripty. Skript je dodán tak, jak je. Otestuje některé základní vlastnosti. Pakliže jsou vaše skripty správné, měl by testovací skript vypsat následující výsledek (vzorové výstupní log soubory najdete v archivu [test.sh.logs.tgz](#)):

```
$ test.sh
smoke test ./le-checkout.sh pass
le-checkout.sh: .le exists pass
le-checkout.sh .le/f1 pass
le-checkout.sh .le/f2 pass
le-checkout.sh ./f1 pass
le-checkout.sh ./f2 pass
le-checkout.sh .le/.config exists pass
le-checkout.sh projdir in .le/.config pass
smoke test ./le-update.sh pass
le-update.sh: modified pass
le-update.sh: updated pass
le-update.sh: unchanged pass
le-update.sh: modified, patched pass
le-update.sh: modified, conflict pass
le-update.sh: local temporary files pass
smoke test ./le-revert.sh pass
le-revert.sh: reverted pass
le-revert.sh: ignored by .config pass
smoke test ./le-diff.sh pass
le-diff.sh: unified diff pass
le-diff.sh: local temporary files pass
```

Last modified: 2013-02-22 10:00