

IOS	public	[http://www.fit.vutbr.cz/study/courses/IOS/public]		Lab
[http://www.fit.vutbr.cz/study/courses/IOS/public/Lab/]				Projekt
[http://www.fit.vutbr.cz/study/courses/IOS/public/Lab/projekt1/]				1

SUSv3 shell+utils [<http://www.opengroup.org/onlinepubs/009695399/utilities/contents.html>]

1. úkol 2011/2012

Naprogramujte skript, který pracuje se sadou regresních testů. Při spuštění s neplatnými parametry (nesprávný přepínač, špatný počet operandů apod.) skript vypíše následující nápovědu na standardní chybový výstup a skončí s návratovým kódem 2 (\$0 je skutečný název spuštěného skriptu, např. `./runtest.sh`, `/usr/local/bin/runtest.sh` nebo `renamed_runtest_in_path.sh`):

```
Usage: $0 [-vtrsc] TEST_DIR [REGEX]
```

```
-v  validate tree
-t  run tests
-r  report results
-s  synchronize expected results
-c  clear generated files
```

```
It is mandatory to supply at least one option.
```

Zpracování přepínačů

- Zpracování přepínačů zajišťuje utilita `getopts(1)` [<http://pubs.opengroup.org/onlinepubs/000095399/utilities/getopts.html>] obvyklým způsobem.
- Přepínače mohou být zadány v libovolné kombinaci, četnosti a pořadí.
- Alespoň jeden přepínač musí být zadán (jinak návratový kód 2 + nápověda).
- Zadané přepínače určují, které jednotlivé operace se provedou.
- Každá operace se provede nanejvýš jednou (i pokud se přepínač opakuje).
- Jednotlivé operace se provedou v pořadí nezávislém na pořadí přepínačů a to následovně: `-v`, `-t`, `-r`, `-s`, `-c`

Zpracování operandů

- `TEST_DIR` udává adresář, nad jehož stromem skript pracuje (dále jen strom).
- Volitelný operand `REGEX` udává rozšířený regulární výraz [http://pubs.opengroup.org/onlinepubs/009604499/basedefs/xbd_chap09.html#tag_09_04] pro filtrování adresářů, se kterými jednotlivé operace pracují (použijte `grep -E` nebo nástroj se stejnou syntaxí regulárních výrazů jako `grep -E`).
- Všechny operace pracují nad adresáři uvnitř `TEST_DIR`, které jsou zadané rozšířeným regulárním výrazem `REGEX`. Pokud volitelný operand `REGEX` není zadán, operace pracují nad všemi adresáři uvnitř `TEST_DIR`.
- Jednotlivé adresáře ve stromu jsou vždy zpracovávány v abecedním pořadí (POSIX lokalizace) podle jejich absolutní cesty.
- Tato pravidla platí pro všechny operace.

Operace -v

- Kontroluje obsah stromu (`TEST_DIR`) vymezeného daným filtrem (`REGEX`).
- Pro každou nesplněnou vlastnost je vypsáno varování.
- Pokud není splněna nějaká vlastnost, návratový kód je nastaven na 1.
- Vlastnosti, které operace `-v` kontroluje, jsou následující:
 - Pokud je v některém adresáři aspoň jeden adresář, tak jsou v něm jenom adresáře a žádné jiné soubory.
 - Ve stromu nejsou žádné symbolické odkazy a žádné vícenásobné pevné odkazy.
 - V každém adresáři, ve kterém nejsou adresáře, existuje soubor `cmd-given` a uživatel má oprávnění jej spouštět.
 - Všechny soubory `stdin-given` v daném stromu jsou uživateli přístupné pro čtení, pokud existují.
 - Všechny soubory `{stdout, stderr, status}-{expected, captured, delta}` v daném stromu jsou uživateli přístupné pro zápis, pokud existují.
 - Všechny soubory `status-{expected, captured}` obsahují pouze celé číslo zapsané v desítkové soustavě následované znakem konce řádku (`0x0A`).
 - Ve stromu jsou pouze adresáře adresářů a výše vyjmenované soubory (a žádné jiné).

Operace -t

- Pro adresáře z daného stromu, ve kterých se nachází soubor `cmd-given`, proved':
 1. Aktuální adresář je nastaven na právě zpracovávaný adresář.
 2. Soubor `cmd-given` z aktuálního adresáře je spuštěn.
 3. Na standardní vstup je spuštěnému souboru předám obsah souboru `stdin-given` z aktuálního adresáře, pokud soubor existuje. Pokud soubor `stdin-given` neexistuje, je na vstup přesměrován soubor `/dev/null`.
 4. Standardní výstup a standardní chybový výstup produkované spuštěným programem jsou zapisovány do souborů `{stdout, stderr}-captured` v aktuálním adresáři.
 5. Návratový kód spuštěného souboru je zapsán do souboru `status-captured` v aktuálním adresáři.
 6. Pro dvojice souborů `X-{expected, captured}` je jejich rozdíl (získaný příkazem `diff -up`) zapsán do odpovídajícího souboru `X-delta`, pro `X` z množiny `{stdout, stderr, status}`.
 7. Na standardní chybový výstup je zapsáno "`$TEST: $RESULT\n`", kde:
 - `$TEST` je cesta adresáře obsahující `cmd-given` relativní k `TEST_DIR` (v kanonickém tvaru bez prefixu `./`).
 - `$RESULT` je `OK`, pokud jsou soubory `{stdout, stderr, status}-delta` prázdné; v opačném případě `FAILED`.
 8. Návratový kód je nastavený na 0, pokud je výsledkem všech testů `OK`.
 9. Pokud je standardní chybový výstup připojen na terminál, jsou řetězce `OK` vypisovány zeleně a řetězce `FAILED` vypisovány červeně.

Operace -r

- Viz operace `-t`, ale body 1 až 5 jsou vynechány a pro výpis výsledků se místo standardního chybového výstupu používá standardní výstup (při tisku i při detekci terminálu).

Operace -s

- Všechny soubory `{stdout, stderr, status}-captured` v daném stromu přejmenuje na odpovídající soubory `{stdout, stderr, status}-expected`.
- Pokud soubory `{stdout, stderr, status}-expected` již existují, jsou přepsány.

Operace -c

- Smaže všechny soubory `{stdout, stderr, status}-{captured, delta}` v daném stromu podle daného filtru.

Návratové kódy

- 0 ... Všechno proběhlo úspěšně.
- 1 ... Požadovaná operace selhala (selhání testu, nevalidní strom, ...).
- 2 ... Špatně zadané parametry, interní chyba skriptu, ...
- Bude-li skript vracet nesprávné návratové kódy, je možné, že přijdete o značnou část bodů vzhledem ke způsobu, jakým budeme projekty opravovat a hodnotit.

Podmínky implementace

- Veškerý kód je obsažen v souboru s názvem `runtests.sh`.
- Skript nepoužívá žádné dočasné soubory.
- Skript nevytváří jiné soubory, než které jsou vyjmenované v zadání.
- Skript nezapíše nic mimo strom daný parametrem `TEST_DIR` (jinak 0 bodů).
- Chybová hlášení a varování jsou vypisována na standardní chybový výstup.
- Přesný obsah chybových a varovných hlášení není specifikovaný. Ve vlastním zájmu vypisujte dostatečně detailní hlášení. Usnadní vám to ladění skriptu.
- Na standardní výstup není vypisováno nic, co není explicitně vyžadováno v zadání.
- Skript musí podporovat zotavení se z chyby (chybějící soubor, nedostatečné oprávnění, nedostatek místa na disku, ...).
- Skript musí fungovat na serverech `{eva, merlin}`.
- První řádek skriptu (tzv. shebang) je `#!/bin/bash` - Pozor! Na serveru `eva` neexistuje `/bin/bash`, tam můžete skript vykonat pomocí explicitního spuštění shellu: `bash ./runtest.sh ...`
- Není dovoleno používat nástroje a/nebo přepínače, které nejsou zahrnuty ve specifikaci POSIX (výjimkou jsou pouze případy uvedené v zadání a případy explicitně povolené opravujícími na diskusním fóru).
- Je dovoleno používat GNU rozšíření `find(1)`, která fungují na serverech `{eva, merlin}`.
- POSIX lokalizaci je možné zapnout pomocí `export LC_ALL=C` na začátku skriptu.
- Můžete předpokládat, že v testovacích stromech se skryté soubory vyskytovat nebudou.

Podmínky odevzdání

- Soubor `runtests.sh` používá kódování 7-bit ASCII, zalamování řádků `0x0A`.
- Soubor `runtests.sh` má nastavený mód `0755`.
- Soubor `runtests.sh` je zabalený pomocí `tar/gzip` jako `xlogin00.tar.gz`.
- Archiv `xlogin00.tar.gz` neobsahuje žádné jiné soubory ani adresáře (`xlogin00` nahradte svým uživatelským jménem na serveru `eva`).
- Archiv odevzdejte pomocí informačního systému v dostatečném předstihu (odevzdaný soubor můžete před vypršením termínu snadno nahradit jeho novější verzí, kdykoliv budete potřebovat).

Doporučené nástroje

- Vestavěné příkazy shellu: `case`, `cd`, `eval`, `exit`, `for`, `getopts`, `printf`, `shift`, `test`, `while`, ...
- Nástroje dostupné na POSIX systémech: `basename`, `cut`, `diff`, `dirname`, `find`, `grep`, `mv`, `readlink`, `sort`, `tty`, `wc`, ...
- Barvený výstup na terminál: escape sekvence [<http://tldp.org/HOWTO/Bash-Prompt-HOWTO/x329.html>], `tput`, ...
- Není nutné využít všechny výše uvedené nástroje.

Časté chyby

- U příkazů, jako je `cd`, není kontrolován návratový kód.
- Konstrukce jako `find "$DIR/" -delete` nebo `rm -rf "$DIR/"`, kde proměnná `DIR` může být prázdná nebo nedefinovaná.
- Špatný nebo chybějící quoting (skript např. nepředpokládá názvy souborů, ve kterých je mezera, dolar, uvozovka, zpětné lomítko, atd.).
- Skript předpokládá konkrétní umístění operandů a/nebo sebe samého (např. funguje `./runtests.sh -r ./tree`, ale nefunguje `cp -v ./runtests.sh /tmp && /tmp/runtests.sh -r ./tree`).
- Duplicitní kód místo používání funkcí, cyklů, atd.
- Do informačního systému je odevzdávána až konečná verze projektu těsně před vypršením termínu (doporučujeme ukládat na server průběžně i částečné verze řešení pro případ, že se vám na poslední chvíli rozbije počítač nebo připojení k internetu).

Příklad použití skriptu

```
$ curl http://www.fit.vutbr.cz/study/courses/IOS/public/Lab/projekt1/tree.tar | tar xf -
$ ./runtests.sh ./tree >/dev/null; printf "EC = %d\n" $?
Usage: ./runtests.sh [-vtrsc] TEST_DIR [regex]

-v validate tree
-t run tests
-r report results
-s synchronize expected results
-c clear generated files

It is mandatory to supply at least one option.
EC = 2
```

```
$ ./runtests.sh -v tree; printf "EC = %d\n" $?
./runtests.sh: directories and other files mixed in: .
./runtests.sh: extra line in ./tests/sort/status-expected
./runtests.sh: not a number in ./tests/sort/status-expected
EC = 1

$ ./runtests.sh -v tree trivial; printf "EC = %d\n" $?
EC = 0

$ ./runtests.sh -st ./tree/tests sort 2>/dev/null; printf "EC = %d\n" $?
EC = 1

$ (./runtests.sh -t ./tree/tests 'sort|trivial/[^k]'; printf "EC = %d\n" $?) | cat
sort: [1;32mOK[0m
trivial/false: [1;32mOK[0m
trivial/true: [1;32mOK[0m
EC = 0

$ ./runtests.sh -crtv ./tree/tests sort >/dev/null; printf "EC = %d\n" $?
sort: [1;32mOK[0m
EC = 0

$ (./runtests.sh -r -t ./tree/tests; printf "EC = %d\n" $?) | cat
expensive/sleep/0.00: [1;32mOK[0m
expensive/sleep/0.25: [1;32mOK[0m
expensive/sleep/0.50: [1;32mOK[0m
expensive/sleep/0.75: [1;32mOK[0m
expensive/sleep/1.00: [1;32mOK[0m
expensive/sleep/1.25: [1;32mOK[0m
expensive/sleep/1.50: [1;32mOK[0m
expensive/sleep/1.75: [1;32mOK[0m
expensive/sleep/2.00: [1;32mOK[0m
sort: [1;32mOK[0m
trivial/false: [1;32mOK[0m
trivial/known-to-fail: [1;31mFAILED[0m
trivial/true: [1;32mOK[0m
expensive/sleep/0.00: OK
expensive/sleep/0.25: OK
expensive/sleep/0.50: OK
expensive/sleep/0.75: OK
expensive/sleep/1.00: OK
expensive/sleep/1.25: OK
expensive/sleep/1.50: OK
expensive/sleep/1.75: OK
expensive/sleep/2.00: OK
sort: OK
trivial/false: OK
trivial/known-to-fail: FAILED
trivial/true: OK
EC = 1
```