Laboratorium/miniprojekt „użytkownicy”

Należy wykonać zadanie podane na stronie przedmiotu. Skrypt może być napisany w Perlu lub języku powłoki. Skrypt zostanie oceniony na 0-7 punktów.

Dodatkowo należy wykonać jedno z zadań z poniższej listy. Numer zadania to ostatnia cyfra numeru indeksu. Skrypt zostanie oceniony na 0-8 punktów.

Wymagania:

- Skrypt musi być napisany w języku powoki bash z użyciem standardowych narzędzi, takich jak tr, awk, grep, sed, itp. Nie należy używać Perla.
- Skrócony opis sposobu wywołania skryptu powinien być dostępny poprzez podanie parametru --help.
- Wymagane jest pełne sprawdzenie poprawności parametrów, istnienia plików wejściowych, itp.
- Wyniki działania skryptu powinny być wyprowadzane na wyjście standardowe lub do pliku, zależnie od opcji -f (np. -f nazwa_pliku lub -f -).

Oddanie obu skryptów powinno nastąpić w trakcie zajęć dn. 27.05.2010.

Projekt 0

Konwerter tekstu podanego jako argument na alfabet Braille'a. W zależności od opcji należy wyświetlać znaki w formie dużej (litery o i spacje, 3 wiersze) lub małej (apostrof, kropka, dwukropek i spacja, dwa wiersze). Pod każdym znakiem alfabetu Braille'a powinna być wyświetlona odpowiadająca mu litera. Np.:

```
$./braille.sh -s tekst
 . .  .   .  .
:'  ' .' :  :'
t  e  k  s  t
```

Projekt 1

Napisać skrypt umożliwiający grę w kółko i krzyżyk między dwoma graczami. Skrypt pyta, kto ma pierwszy ruch, po czym cyklicznie wyświetla planszę i pyta użytkownika o kolejne posunięcie, np:

```
   1   2   3
1  o |   |
  ---+---+---
2  x | x | o
  ---+---+---
3    |   |

Gracz x - Nastepny ruch (wiersz, kolumna)?
```

Projekt 2

Odwracalne usuwanie plików. Pliki podane w linii poleceń powinny być kompresowane i przenoszone do katalogu $HOME/.smietnik. Pliki już skompresowane nie powinny być kompresowane ponownie (polecenie file umożliwia ich identyfikację). Przy każdym wywołaniu skrypt powinien usuwać ze śmietnika pliki starsze, niż 48 godzin. Skrypt powinien obsługiwać usuwanie zarówno pojedynczych plików, jak i całych drzew katalogów.

Projekt 3
Uniwersalny skrypt do dekompresji plików skompresowanych. Skryptpowinien identyfikować przy użyciu polecenia file narzędzię którym dany plik został skompresowany, a następnie wykorzystać to narzędzie do jego rozpakowania. Skrypt powinien umożliwiać – zależnie od parametrów – rozpakowanie jednego pliku, wielu plików, a także plików z listy podanej w pliku tekstowym, a także wszystkich plików w katalogu i jego podkatalogach.

Projekt 4
Graficzne (grafika tekstowa) wyświetlanie drzewa katalogów, którego korzeń podano jako parametr wejściowy. Wyświetlanie obejmuje tylko katalogi, zwykłe pliki należy pomijać. W zależności od opcji wywołania skrypt powinien podążać lub nie za łączami symbolicznymi do katalogów, o ile nie wskazują one na katalog znajdujący się już w drzewie (w takim przypadku należy wyświetlić nazwę łącza i nazwę katalogu docelowego. Skrypt powinien być napisany przy użyciu standardowych narzędzi typu sed, grep, find itp.

Projekt 5
Skrypt do zarządzania chronologiczną kolekcją plików. Skrypt powinien potrafić konwertować kolekcję między dwoma formatami, oraz zaimportować (na podstawie dat utworzenia pliku) dowolny katalog lub listę katalogów do kolekcji o wybranym formacie. Kolekcja jest katalogiem, którego zawartość ma określoną strukturę.
Format 1: Pierwszy poziom podkatalogów nazwany jest datami wformacie rrrr.mm.dd. Zawartość każdego z podkatalogów stanowią pliki utworzone w danym dniu. Pliki te powinny się znajdować w strukturze podkatalogów zgodnej z oryginalną.
Format 2: Podkatalogi są zgodne z oryginalną strukturą, ale pliki w nich zawarte są umieszczone w podkatalogach o nazwach rrrr.mm.dd.
W przypadku wystąpienia duplikatów przy dodawaniu plików do kolekcji, należy sprawdzać, czy oba pliki są takie same. Jeśli tak, należy zachować po prostu jedną kopię, jeśli nie, zapytać użytkownika o wybór wersji lub pominięcia pliku. Skrypt powinien móc tworzyć plik z listą plików pominiętych przy imporcie, aby umożliwić późniejsze ręczne sprawdzenie tych przypadków.
Przykład struktury katalogów w obu formatach, utworzonej na podstawie katalogu z plikami z dwóch różnych dni:

Oryginalny	Format 1	Format 2
/a/plik1	/2010.05.11/a/plik1	/a/2010.05.11/plik1
/a/plik2	/2010.05.11/b/plik3	/a/2010.05.12/plik2
/b/plik3	/2010.05.11/plik5	/b/2010.05.11/plik3
/c/plik4	/2010.05.12/a/plik2	/c/2010.05.12/plik4
/plik5	/2010.05.12/c/plik4	/2010.05.12/plik5

Projekt 6
Skrypt do tworzenia miniaturek plików graficznych i filmów. Skrypt powinien przy pomocy polecenia file sprawdzać, czy podane na wejściu pliki są odpowiedniego typu, a jeśli podano katalog, wyszukiwaćw nim i jego podkatalogach takie pliki. Dla każdego z tych plików należy utworzyć miniaturkę, zachowującą jego proporcje, ale mieszczącą się w prostokącie o podanych wymiarach. Nazwa miniaturki powinna różnić się od oryginału przyrostkiem _short bezpośrednio przed rozszerzeniem (jeśli nazwa ma rozszerzenie), oraz ewentualnie rozszerzeniem. Należy wykorzystać pakiety imagemagic i ffmpeg.

Projekt 7
Konwerter plików tekstowych na HTML. Plik wejściowy przestrzego następujących konwencji:

- paragrafy oddzielone pustą linią,
- obrazy są umieszczone w katalogu images, odwołania do nich pojawiają się w pliku tekstowym w nawiasach kwadratowych,
- tekst otoczony podkreśleniami (ze spacjami, czyli " _" i "_ ") powinien być wyświetlany czcionką pochyłą,
- tekst otoczony gwiazdkami (ze spacjami, czyli " *" i "* ") powinien być wyświetlany czcionką wytłuszczoną,
- tekst w nawiasach klamrowych jest łączem lub parą tekst|łącze (pionowa kreska), np. {http://example.com} lub {przykładowe łącze|http://example.com}.

Projekt 8
Konwerter plików HTML na tekstowe. Program powinien działać odwrotnie do opisanego w projekcie 7, czyli odpowiednio dekodować znaczniki b, i, img, a, p i br. Można założyć, że plik wejściowy jest wynikiem działania skryptu z projektu 7 (czyli nie ma komentarzy, itp.).

Projekt 9
Napisać skrypt wyświetlający zadany ciąg znaków w formie zygzaka o podanej długości fazy opadającej i wznoszącej. Większa z tych wartości + 1 określa wysokość zygzaka. Np:

```
$./zygzak.sh 6 3 "To jest tekst testowy, zróbmy z niego zygzak"
T        e        o        m        o
 o        k        w        y
        t  s     t  y     b        g  z
   j        t        ,        z        y
    e           s        ó        e     g  k
     s        t        z        n        z
      t        e        r        i        a
```