

Projekt zaliczeniowy 2023

Przygotuj skrypt, który dokona analizy i rozdzielania danych na inne pliki

Skrypt powinien mieć nazwę poprawka2.sh oraz powinien dać się uruchomić w środowisku testowym.

Uruchomienie skryptu będzie z następującymi parametrami:

`./poprawka2.sh <KAT_BAZOWY> <nazwa_pliku_danych1> <nazwa_pliku_danych2> ...`

`<nazwa_pliku_danychN>`

Gdzie:

`<KAT_BAZOWY>` - nazwa katalogu bazowego wykorzystywanego do tworzenia docelowej struktury danych

`<nazwa_pliku_danych1..N>` - nazwa pliku z danymi do analizy, ilość plików dowolna

Skrypt powinien sprawdzić poprawność uruchomienia (np. czy został uruchomiony poprawnie, oraz przed rozpoczęciem działania sprawdzić czy pliki danych istnieją, itp.). W przypadku błędów na standardowy strumień błędów ma być wysłany stosowny komunikat oraz skrypt powinien zakończyć pracę z odpowiednim kodem błędu.

Kody błędów to:

- 1 - brak argumentów, komunikat z błędem wypisuje poprawy sposób uruchomienia,
- 2 - tylko jeden argument
- 3 - brak dostępu do pliku danych lub plik nie istnieje, komunikat wypisuje nazwę pliku którego dotyczy komunikat
- 4 - brak możliwości utworzenia katalogu bazowego

Kod błędu jest zwracany przez polecenie "exit N", gdzie N to kod błędu.

Działanie skryptu podzielone jest na zadania. Przed każdym zadaniem należy umieścić numer zadania jako wiersz:

`#NR_ZAD START`

a na końcu

`#NR_ZAD STOP`

Przykładowo:

`#1 START`

... tutaj komendy do zadania 1 ...

`#1 STOP`

`#2 START`

... tutaj komendy do zadania 2 ...

`#2 STOP`

Jeżeli jakiś fragment odpowiada za realizację więcej niż jednego zadania, zamiast jednego numeru podaj listę numerów zadań rozdzielonych przecinkami

Zadania skryptu:

0.

Skrypt dokona pomiaru czasu własnej pracy. Wykona to przez dwukrotny odczyt czasu systemowego i wyliczenia ile milisekund skrypt się wykonywał. Obliczenia wykona z dokładnością nie gorszą niż mikrosekundową a wynik zostanie zapisany do pliku out.log w katalogu bazowym. Format zapisanych danych to:

`PID,PPID,<obliczony_czas>,<wiersz poleceń uruchomienia skryptu>`

Jeżeli plik "out.log" już istnieje, to wynik zostanie dopisany na jego końcu.

1.

Dla zadanych parametrów skrypt utworzy strukturę katalogów wg wzorca

`<KAT_BAZOWY>`

<ROK>

<MIESIAC>

Gdzie:

KAT_BAZOWY - nazwa podana jako parametr wywołania skryptu

ROK, i MIESIAC - to rok i miesiąc rejestrowanych danych. Nazwy katalogów mają być utworzone na podstawie danych jakie są plikach, czyli np. plik z danymi zawiera pomiary z roku 2022 i pierwszego kwartału, nazwa katalogu bazowego to "dane_opadowe" wtedy utworzone katalogi będą:

dane_opadowe

2022

01

02

03

Jeżeli katalogi już istnieją to skrypt nie będzie zgłaszał błędów.

Wszystkie utworzone katalogi mają atrybuty ustawione tak aby właściciel mógł je swobodnie przeglądać i modyfikować a grupa tylko czytać i przeglądać a pozostali nie mieli dostępu (przeglądać = wykonać polecenie cd nazwa_katalogu).

Pliki mają mieć możliwość zapisu tylko przez właściciela, a odczyt tylko przez właściciela i grupę.

2.

Dane z plików podanych jako argumenty zostaną rozdzielone wierszami(skopiowane) do odpowiednich plików w odpowiednich katalogach. Nazwa tworzonych plików jest wg wzorca:

DD.csv

Gdzie:

DD - numer dnia miesiąca rejestracji danych.

Każdy wiersz z danych wejściowych zostanie skopiowany do odpowiedniego pliku i katalogu zgodnie z datą rejestracji.

Kopiowaniu podlegają tylko wiersze dla których "status pomiaru SMDB" jest różny od "8" (patrz opis pliku danych).

3.

Wiersze z plików dla których "status pomiaru SMDB" jest równy "8" zostaną skopiowane do pliku o nazwie

<ROK>.<MIESIAC>.errors umieszczonym w katalogu bazowym.

4.

Dodatkowo w katalogu bazowym utworzyć katalog LINKS, a w nim linki symboliczne o nazwie MIN_OPAD i MAX_OPAD do plików w których suma dzienna opadów jest najmniejsza oraz największa ze wszystkich analizowanych dni. Jeżeli więcej niż jeden plik ma wartość równą maksymalnej lub minimalnej, wystarczy wybrać dowolny z nich (w przykładowych danych chyba nie ma takiej sytuacji). Linki nie mogą być w postaci bezwzględnej, czyli bazować na katalogu "/" lub "~".

Opis formatu plików danych jest w pliku o_d_format.txt.

Przykładowe dane pochodzą ze stron IMGW, poddane zostały tylko lekkiemu filtrowaniu i selekcji.

Skrypt może tworzyć pliki tymczasowe (jeżeli potrzeba) ale tylko w katalogu bazowym, jednak kończąc pracę musi je wykasować.

Dane w pliku wejściowym, każde pole, może być zapisane w cudzysłowach, ale nie musi. W danych zapisywanych przez skrypt te cudzysłowy powinny zostać usunięte.