

Podstawowy warsztat informatyka — lista 4 — zdalna

W związku z AMPPZ, listę tę należy wykonać samodzielnie oraz wysłać przez odpowiednie zadanie w systemie skos najpóźniej we wtorek, 30 października, o 23.59. Rozwiązanie powinno składać się z jednego pliku tekstowego zawierającego imię i nazwisko autora, nazwę i termin grupy oraz rozwiązania poniższych zadań. Plik powinien być podpisany na początku, a każde rozwiązanie należy poprzedzić wykomentowanym numerem zadania, np.

Jan Kowalski, grupa MML, śr 9:15

Zadanie 1

1. (1 punkt) Wykonaj w terminalu polecenia:

```
sleep 3600 &
nohup sleep 3600 &
sleep 3600
```

Następnie w drugim terminalu wykonaj polecenie `ps -eF | grep ^user`, wpisując zamiast `user` swoją nazwę użytkownika. Co oznacza wynik tego polecenia?

W pierwszym terminalu wciśnij `ctrl+z`, aby zatrzymać proces, a następnie uruchom polecenie `jobs`. Wznów zatrzymane polecenie komendą `bg`, a następnie `fg`. Zabij to polecenie skrótem `ctrl+c`. Sprawdź powtarzając polecenie w drugim terminalu, czy rzeczywiście umarło.

W pliku z rozwiązaniem należy własnymi słowami opisać, czym się różni `sleep 3600 &` od `nohup sleep 3600 &` oraz jaki był efekt wykonania poleceń `bg`, `fg`.

2. (1 punkt) Znajdź w katalogu `/usr/share` wszystkie pliki, które nie zawierają słowa `good` i których nazwa kończy się na `right`. Wyświetl nazwy tych plików. Następnie zapisz jedną sekwencję poleceń oddzielonych znakiem `|` zawartość wszystkich tych plików do pliku `wszystkie`. Jako rozwiązanie prześlij tę sekwencję. *Wskazówka:* Wygodnie to zrobić poleceniem `xargs`.
3. (2 punkty) Będziemy przetwarzać polskie tłumaczenie *Romea i Julii* (projekt Gutenberg) <http://www.gutenberg.org/files/27062/27062-0.txt>. Do wykonania zadania nie jest potrzebne pisanie jakichkolwiek skryptów, przyda się znajomość opcji programów: `grep`, `sort`, `uniq`. Dla każdego podpunktu prześlij jedną sekwencję poleceń wyliczających pożądaną odpowiedź.
 - (a) (0.5 punkta) Policz ile kwestii wypowiada Julia. Zakładamy że każda kwestia Julii jest oznaczona linijką która zaczyna się od `*Julia.*`. Pamiętaj że `grep` domyślnie traktuje podane wyrażenie jako wyrażenie regularne, a znaki jak gwiazdka i kropka mają wtedy specjalne znaczenie — użyj backslash (albo opcji `-fixed-strings`) żeby się przed tym zabezpieczyć.
 - (b) (0.5 punkta) Policz ile kwestii jest wypowiedzianych w całym dramacie. Jak powyżej, zakładamy że 1 kwestia = 1 linia zaczynająca się od gwiazdki + nazwa postaci (ciąg liter, być może z polskimi ogonkami) + kropka + gwiazdka. Czyli trzeba będzie napisać coś podobnego jak wyżej, tylko teraz nie można już zahardcodować nazwy *Julia*, zamiast tego trzeba napisać wyrażenie regularne które dopuści dowolną nazwę postaci. *Wskazówka:* może przyda się wyrażenie `*.*\.*\.*`
 - (c) (0.5 punkta) Użyj `grep` (z odpowiednią opcją output), `sort`, `uniq` żeby wypisać wszystkie nazwy postaci które wypowiadają jakieś kwestie w dramacie. Wypisane nazwy mogą mieć gwiazdkę na początku, i kropkę+gwiazdkę na końcu. Ważne żeby każdą nazwę postaci wypisać tylko raz (modulo ewidentne błędy edytorskie np. *druugi* vs. *drugi*).
 - (d) (0.5 punkta) Wypisz krótkie podsumowanie: dla każdej postaci ile razy dana postać wypowiada kwestię w dramacie, a następnie nazwa tej postaci (z ew. gwiazdką i gwiazdką+kropką naokoło). Postacie posortuj wg liczby wypowiedzianych kwestii zaczynając od największej. Moje rozwiązanie zaczyna się tak:

160 *Romeo.*
111 *Julia.*
89 *Marta.*
61 *Merkucio.*
61 *Benwolio.*

Ostatnie zadanie jest modyfikacją zadania Michalisa Kamburelisa.