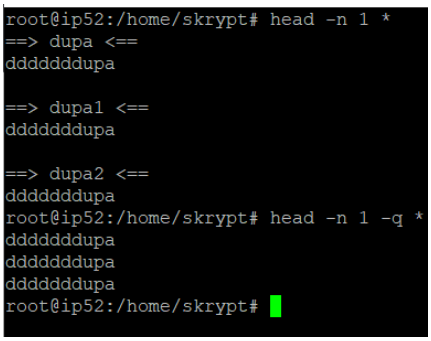
Systemy Operacyjne

Wejściówka 2

Lab_3 + Lab_4 + Lab_5

1. Strumienie wejścia wyjścia i dodawania do pliku:
    - > (strumień wejścia, wrzuca dane do pliku i nadpisuje plik)
    - < (strumień wyjścia przekazuje zawartość pliku)
    - >> (strumień wejścia – działa jak > ale dopisuje na końcu pliku NIE USUWA)
2. Polecenie **cut** służy do „wycinania / wyciągania” części z pliku:
        - **-c** (dla określonych znaków)
        - **-d** (dla określonych separatorów np. spacja, tabulator, dwukropek) **(-d " " -> dla spacji)**
        - **-f** (dla określonych pól)
    - **cut -c 4-10 plik.txt** Pokazuje nam od 4 do 10 znaku z każdej linii w pliku
    - **cut -d : -f 5 plik.txt** Pokazuje nam 5 pole gdy pola oddzielone są :
    - **cut -f 2 plik.txt** Pokazuje nam 2 pole gdy pola oddzielone są tabulatorem
3. Polecenie **cat** służy do wyświetlania zawartości pliku:
        - **-n** (numerowanie wierszy)
    - cat plik.txt (wyświetla zawartość pliku)
    - cat < plik.txt (wyświetla zawartość pliku)
    - cat > plik.txt (możemy wpisywać do pliku z konsoli)
4. Polecenie **head** służy do wyświetlania początkowej części pliku:
        - **-n** (ilość wierszy od góry)
        - **-q** (usuwanie informacji o pliku, z którego odczytujemy – przydatne gdy wyświetlany na raz kilka plików)
        - 

    - **head -n 5 plik.txt** (wyświetlamy 5 pierwszych wierszy)
5. Polecenie **more** służy do wyświetlania zawartości pliku z możliwością poruszania się po nim. PS tyko w dół XDDDD
    - **more plik.txt**
6. Polecenie **less** to takie **more** ale na dopalaczach**,** możemy skrolować plik:
    - **less plik.txt**
7. Polecenie **tail** to odwrotność **head**, służy do odczytywania końcówki pliku:
        - **-n** (ilość wierszy od dołu)
        - **-f** (gdy chcemy monitorować plik pod kątem zmian - zmienia się na żywo)
        - **-q** (usuwanie informacji o pliku, z którego odczytujemy – przydatne gdy wyświetlany na raz kilka plików)
    - **tail -n 20 plik.txt** (wyświetlanie 20 ostatnich wierszy z pliku)
8. Polecenie **wc** służy do zliczania (wierszy, słów, znaków):
        - **-l** (zlicza wiersze w pliku)
        - **-m** (zlicza liczbę znaków w pliku)
        - **-w** (zlicza liczbę słów w pliku)
    - **wc -l plik.txt**

9. Polecenie **uniq** sprawdza czy nie występują jakieś powtórzenia (**UWAGA** używać tylko gdy posortowaliśmy zawartość):
    - **-c** (wyświetlanie ile razy dany wiersz się powtórzył)
    - **-d** (wyświetlanie tylko powtarzających się wierszy)
    - uniq plik.txt (najlepiej stosować po sortowaniu **sort plik.txt | uniq** można także użyć **cat plik.txt | sort | uniq**)
10. Polecenie **cmp** służy do porównywania zawartości plików (**bajt po bajcie**):
    - **-c** (wyświetla specyfikację znaków)
    - **-l** (powoduje, że polecenie nie zatrzymuje się na pierwszej napotkanej różnicy)
    - **cmp plik1.txt plik2.txt** (pokaże pierwszą różnice w plikach)

```
root@ip52:/home/skrypt# cmp dupa dupa1
dupa dupa1 differ: byte 1, line 1
```

11. Polecenie **diff** również służy do porównywania, ale jest bardziej zaawansowane i pokazuje każdą różnice:
    - **diff plik1.txt plik2.txt**

```
root@ip52:/home/skrypt# diff dupa dupa1
1,3c1,2
< ddddddddupa aaaaaaaaaa
< dskdsffds
< 21433
---
> ddddddddupa
> 1221
root@ip52:/home/skrypt#
```

12. Polecenie **sort** sortuje dane (UWAGA, dopiero po **sort** można używać **uniq**):
    - **-r** (rekursywnie czyli od tyłu)
    - **sort plik.txt**
    - **cat plik.txt | sort** (sortowanie po wyciągnięciu danych z pliku przez cat)
13. Polecenie **tr** zamienia znaki na inne, może usuwać nadmierne ilości znaków np. kropki
    - **'a-z' 'A-Z'** (zmienia małe znaki na duże)
    - **-d '**<znak>**'** (usuwa podany znak)
    - **-s '**<znak>**'** (usuwa nadmierna ilość danego znaku)
    - **cat plik.txt | tr 'a-z' 'A-Z'** (zmieniamy małe na duże litery)
    - **cat plik.txt | tr -d '.'** (usuwamy .)

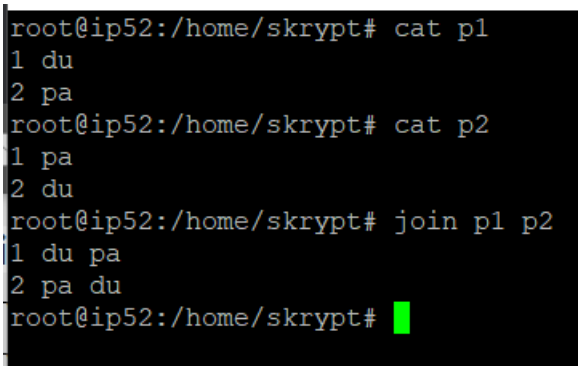
```
root@ip52:/home/skrypt# cat dupa2
ddddddddupa......
root@ip52:/home/skrypt# cat dupa2 | tr -d '.'
ddddddddupa
root@ip52:/home/skrypt#
```

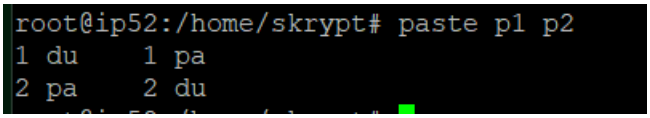
    - **cat plik.txt | tr -s 'd'** (usuwamy nadmiar d)

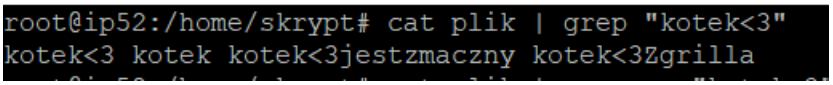
```
root@ip52:/home/skrypt# cat dupa2 | tr -s 'd'
dupa......
root@ip52:/home/skrypt#
```

    - **cat plik.txt | tr "pa" "py"** (zamiana pa na py)

```
root@ip52:/home/skrypt# cat dupa2 | tr "pa" "py"
ddddddddupy......
root@ip52:/home/skrypt#
```

14. Polecenie **join** łączy ze sobą pliki w jeden spójny:
    - **join plik1.txt plik2.txt**
    - 

15. Polecenie **paste** zlepia linie podanych plików oddzielając je znakiem tabulacji:
    - **paste plik1.txt plik2.txt**
    - 

16. Polecenie **grep** wyszukuje podaną przez nas frazę (działa jak sito):
    - -c (zlicza znalezione fragmenty)
    - -w (wyszukuje całe słowa)
    - -v (wyszukuje całe linie)
    - -i (ignoruje wielkość liter)
    - -x (gdy cała linia pasuje)
    - **cat plik.txt | grep kotek<3**
    - **grep "kotek<3" plik.txt**
    - 

    - **ls -l | grep ^-** (pokazanie tylko plików)

---

1. Proces:
    - Każdy uruchomiony w systemie program nosi nazwę procesu, którego składowymi są:
        - dane programu
        - dane systemowe
        - kod binarny procesu załadowany z pliku
    - Do danych systemowych zaliczyć możemy:
        - identyfikator procesu (PID)
        - identyfikator procesu macierzystego (PPID)
        - standardowe strumienie danych
        - środowisko procesu

2. Gdy wylistujemy procesy (**ps aux**) u góry z zobaczymy wiersz z:
    - **USER** - nazwa właściciela
    - **PID** - numer procesu
    - **%CPU** – ilość zużywanego procesora
    - **%MEM** – ilość zajmowanej pamięci RAM
    - **PPID** - numer procesu nadrzędnego
    - **C** - liczba potomków (dzieci)
    - **STIME** - czas uruchomienia
    - **TTY** – terminal w których został uruchomiony proces
    - **TIME** – czas procesora (ile procesor czasu poświęcił na proces)
    - **CMD** - nazwa procesu
3. Zabijanie procesu **kill <pid>**
4. Czym jest priorytet procesu i co to jest parametr NICESNESS?
    - Priorytet procesu: Wartości wynoszą od 0 do 139 gdzie od 0 do 99 czasu rzeczywistego i 100 do 139 dla użytkowników. jest to rzeczywista wartość priorytetu procesy widziana przez jądro Linuxa.
    - Parametr NICESNESS przyjmuje wartości od -20 do 19, czym wyższa wartość tym mniejszy priorytet. Podstawowa wartość to 0. UWAGA tylko użytkownik o uprawnieniach superusera może używać ujemnych wartości parametru niceness.
5. Użytkownik efektywny EUSER a rzeczywisty RUSER, kiedy przechodzimy z rzeczywistego na efektywny:
    - Rzeczywisty użytkownik – ten ogólny co mamy np. zajebistystudent
    - Efektywny użytkownik – zwiększenie uprawnień np. przez sudo, lub gdy zmieniamy hasło