Grep, regexp, pipe i inne, czyli praca na plikach

Poznaliśmy już podstawy pracy na plikach, znaki specjalne, dziś przychodzi moment, aby zająć się tym, co pliki zawierają.

Grep czy egrep, oto jest pytanie.

Polecenie grep, w rozwinięciu *global regular expression print,* jak sama nazwa wskazuje, wydrukuje nam na konsoli to, co mu podamy w wyrażeniu regularnym.

```
grep [opcje] [wyrażenie] [plik]
```

W czym?

W wyrażeniu regularnym, czyli wyrażeniu opisującym jakiś ciąg symboli. Bawiliśmy się trochę upośledzoną wersją tego na poprzednich zajęciach, dziś poznamy wersję dla prawdziwych informatyków

egrep oznacza rozszerzoną wersję grepa, *extended regular expression print.* Ten sam efekt uzyskamy używając zwykłego polecenia

```
grep -E
```

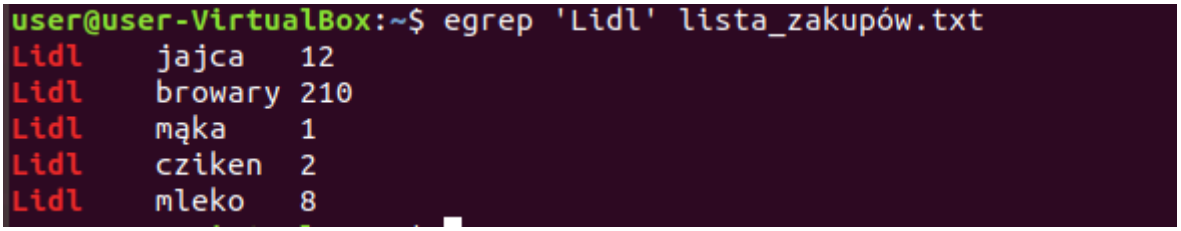
Wyrażenia regularne

Działają na podobnej zasadzie co poznane wcześniej wildcardy, ale elementy, z których budujemy wyrażenia regularne są trochę bardziej złożone, i jest ich znacznie więcej.

- `.` – pojedynczy znak
- `?` – poprzedni znak pojawia się 0 lub 1 raz
- `*` – poprzedni znak pojawia się 0 lub więcej razy
- `+` – poprzedni znak pojawia się 1 lub więcej razy
- `{n}` – poprzedni znak pojawia się n lub więcej razy
- `{n,m}` – poprzedni znak pojawia się od `n` do `m` razy
- `[abc]` – znak jest jednym z wymienionych w nawiasie
- `[^abc]` – znak nie jest jednym z wymienionych w nawiasie
- `()` – pozwala na grupowanie znaków
- | – operacja logiczna OR
- ^ – początek linii
- $ – koniec linii (za koniec przyjmujemy tutaj znacznik LF, windowsowe CRLF może nie zostać wykryte)

Zacznijmy od listy zakupów, dla wygody umieściłem ją na stronie, więc proszę pobrać plik lista_zakupów.txt. Zaczniemy od czegoś prostego.

```
egrep 'Lidl' lista_zakupów.txt
```



Powyższe polecenie wydrukuje nam wszystkie zakupy, które mamy zrobić w Lidlu. Ale przecież część mogliśmy zapisać od małej litery. Żeby wypisać nam również zakupy, które mieliśmy zrobić w **lidlu,** użyć musimy np. nawiasów kwadratowych.

```
egrep '[lL]idl' lista_zakupów.txt
```

```
user@user-VirtualBox:~$ egrep '[Ll]idl' lista_zakupów.txt
Lidl    jajca   12
Lidl    browary 210
Lidl    mąka    1
lidl    kakao   7
Lidl    cziken  2
lidl    boczek  2
Lidl    mleko   8
```

Teraz wyszukamy produkty, które chcemy kupić w dwóch sztukach. Jednak gdy użyjemy polecenia

```
egrep '2' lista_zakupów.txt
```

Dostaniemy wszystkie pozycje, które zawierają cyfrę '2', a nie o to mi chodziło. Przyjść mogłoby nam z pomocą dodanie znaku końca linii po '2'

```
egrep '2$' lista_zakupów.txt
```

ale w tym wypadku dostaniemy również jajca, które chcemy kupić w ilości sztuk 12. W tym wypadku pomocne nam będą tzw. shorthands, które określają nam konkretne klasy znaków

- \w – określa znaki słów, czyli z zakresu [A-Za-z0-9_]
- \s – określa znaki białe, takie jak spacje, tabulatory
- ~~\d – określa cyfry 0-9~~ - grep od wersji 3.4 nie wspiera tego skrótu, zamiast tego używamy [0-9] lub [[:digit:]]

czyli w wypadku, w którym chcemy wypisać wszystkie pozycje, których zakupić chcemy 2 sztuki, najlepiej będzie użyć polecenia

```
egrep '\s2$' lista_zakupów.txt
```

```
user@user-VirtualBox:~$ egrep '\s2$' lista_zakupów.txt
Lidl    cziken  2
lidl    boczek  2
Auchan  brokuł  2
Auchan  proszek 2
Żabka   hotdog  2
```

Aby w pliku lista_zakupów.txt znaleźć wszystkie produkty, które zamierzamy kupić w Lidlu bądź Auchanie w ilości 2 użyjemy polecenia

```
egrep '([lL]idl|Auchan)\s\w*\s2$' lista_zakupów.txt
```

Aczkolwiek da się to zrobić prościej. Jak ktoś ma pomysł, to zapraszam do tablicy, oferuję kolejnego nic nieznaczącego plusika

Zadanie

W zakoszonym ze strony CKE pliku komputery.txt znajdują się trzy kolumny oddzielone znakiem tab. Pierwsza to numer komputera, druga to literowe oznaczenie sekcji, w której ten komputer się znajduje, trzecia natomiast oznacza pojemność dysku danego komputera, dla ułatwienia trzycyfrowa. Używając grepa wyszukaj komputery znajdujące się w sekcji R, N lub Z, które mają pojemność dysku >=600.

Piping & redirection

Wrócimy na chwilę do całkowitych podstaw. Co się wydarzy, gdy zastosujemy polecenie ls? Na konsoli wydrukuje nam się lista plików i katalogów. Ale przypomnijmy sobie inne polecenie z pierwszych zajęć:

```
history > history.txt
```

To polecenie powodowało zapisanie do pliku history.txt całej historii terminala. Pomiędzy poleceniem history, którym, swoją droga można sobie konkretnie zaśmiecić terminal, a plikiem, do którego tę historię zapisujemy, stoi znak większości >. Ten znak odpowiada za przekazanie do podanego po prawej stronie wyrażenia pliku wyniku działania polecenia po lewej. Tak więc, aby zapisać do pliku wynik polecenia ls użyjemy polecenia

```
ls > lista.txt
```

```
user@user-VirtualBox:~$ ls > pliki.txt
user@user-VirtualBox:~$ cat pliki.txt
Dane_PR2
Desktop
Documents
Downloads
JSON.txt
komputery.txt
lista.txt
lista_zakupów.txt
Music
Pictures
pliki.txt
Public
Templates
Videos
user@user-VirtualBox:~$
```

Na identycznej zasadzie zadziała nam zapis do pliku z chociażby wspomnianego wcześniej grepa.

```
user@user-VirtualBox:~$ egrep '(.idl|Auchan).*\s2$' lista_zakupów.txt
Lidl    cziken  2
lidl    boczek  2
Auchan  brokuł  2
Auchan  proszek 2
user@user-VirtualBox:~$ egrep '(.idl|Auchan).*\s2$' lista_zakupów.txt > zakupy.
txt
user@user-VirtualBox:~$ cat zakupy.txt
Lidl    cziken  2
lidl    boczek  2
Auchan  brokuł  2
Auchan  proszek 2
```

Zasada działania jest wyjątkowo prosta. Każdy program (polecenie) ma trzy strumienie danych. Przetwarza dane wejściowe (STDIN) na dane wyjściowe (STDOUT), które domyślnie drukuje na terminalu, ewentualnie drukuje na nim dane o błędach (STDERR). Piping i redirection przekierowują nam któreś z tych trzech strumieni danych w inne miejsce.

- > - zapisuje output do pliku
- >> - dodaje output do istniejącego pliku
- | - przekierowuje output do innego programu
- < - wczytuje treść pliku na input
- 2> - zapisuje błędy do pliku.

Więc, dla przykładu, zapiszemy teraz do pliku zabka.txt ponumerowaną listę zakupów z żabki. Żeby to osiągnąć musimy najpierw zastosować grepa, a następnie wynik jego działania przekazać do nl. po tej czynności możemy już zapisać dane wyjściowe do pliku.

```
egrep 'Żabka' lista_zakupow.txt | nl > zabka.txt
```

```
user@user-VirtualBox:~$ cat zakupy_zabka.txt
     1  Żabka   hotdog  2
     2  Żabka   cola    0
     3  Żabka   kawa    1
     4  Żabka   serek   1
user@user-VirtualBox:~$
```

Zadanie

Z pliku komputery.txt znajdź wszystkie komputery z pierwszych 100 i ostatnich 150 linii (przydać się mogą polecenia z ubiegłego tygodnia), o oznaczeniu sekcji R, N lub Z, ponumeruj je i zapisz do pliku PC.txt.