

	Państwowa Wyższa Szkoła Zawodowa w Nysie		Wydział Nauk Technicznych		
	Laboratorium Podstaw Systemów Komputerowych				
Kierunek:	Informatyka	Rok studiów nr:	1	Semestr nr:	2
Rok akademicki:	2020/2021	Grupa administracyjna:	L5	Grupa ćwiczeniowa:	L5g1

## SPRAWOZDANIE

Nr ćwiczenia	Temat ćwiczenia			
3b	Strumienie standardowe, filtry i potoki – część II			
Termin złożenia sprawozdania				
Termin wg listy				
Data faktycznego złożenia sprawozdania				
(nie wypełniaj)				
Wykonawcy	Nazwisko	Imię	Nr indeksu	Ocena
	Roszak	Damian		(Nie wypełniane w trybie online)
				(Nie wypełniane w trybie online)

**Uwaga:** Umieszczenie danych osobowych wykonawców stanowi grupowe i nieodwołalne oświadczenie, że są oni/one (i tylko oni/one) współautorami przedstawionego sprawozdania. Późniejsza zmiana składu zespołu wykonawców nie będzie możliwa.

Nie wypełniać przy składaniu online

Data i podpis prowadzącego  
ćwiczenia

### Wymagania typograficzne

- Tekst główny (w ramach) należy składać czcionką normalną typu **Times 12 pkt.**
- Zawartość plików, nazwy ścieżek w systemie plików, polecenia wydawane z konsoli i uzyskiwane odpowiedzi systemu/aplikacji oraz kopie tabulogramów interakcji z powłoką należy składać czcionką normalną typu **Courier 11 pkt.** Należy zachować wygląd, w tym pozycjonowanie tekstu.
- Nazwy pozycji menu w programach i nazwy przycisków ekranowych należy składać czcionką pogrubioną typu **Arial 11 pkt.**
- Wykluczone jest zamieszczanie ilustracji graficznych z ciemnym tłem. Tekst powinien z tłem wyraźnie kontrastować.

## 1. Temat ćwiczenia

(kopia tematu instrukcji, identyczna jak tytuł sprawozdania)

### Strumień standardowe, filtry i potoki – część II

## 2. Zakres ćwiczenia

Streszczenie treści ćwiczenia oraz ustalenia prowadzącego zajęcia dotyczące wyboru funkcji badanego programu, zastosowanego algorytmu, zbioru przetwarzanych danych, precyzji przedstawienia liczb, liczby wątków i cykli obliczeń, sposobu prezentacji wyników, itp.)

Przedmiotem ćwiczenia jest:

- 1) zapoznanie się z koncepcją filtru i poznanie kilku wybranych filtrów,
- 2) zapoznanie się z koncepcją potoku i przykładami zastosowań potoków.

## 3. Środowisko realizacji ćwiczenia

(architektura logiczna systemu – sprzęt, elementy składowe, ich cechy i sposób wzajemnego połączenia, schematy; wykorzystywane języki, oprogramowanie, biblioteki, skrypty powłokowe, zasoby sieciowe i dokumentacja)

## 4. Przebieg ćwiczenia i uzyskane wyniki

(przedstawienie czynności wykonanych w ramach realizacji ćwiczenia, w kolejności określonej treścią instrukcji.

Dla każdego punktu instrukcji należy przedstawić: nr i tytuł tego punktu, cel działania, sposób wykonania, otrzymany rezultat i jego ocenę). Wymagana jest 100% chronologia zadań, czynności i uzyskanych rezultatów.

### 4.1 Zadanie nr 1

4.1.1 Nr i treść polecenia wg instrukcji

4.1.2 Cel czynności

4.1.3 Sposób i rezultat wykonania polecenia (np. polecenia wydane na konsoli i odpowiedź systemu/aplikacji, w postaci wycinka zarejestrowanego logu konwersacji terminalowej w formacie tekstowym). Dopuszcza się zamieszczenie fragmentu zrzutu ekranowego. W każdym przypadku obraz rezultatu ma obejmować wykonania wyłącznie danego punktu (a nie wszystko, co widać w oknie terminala lub konsoli). Log konwersacji musi zawierać następujące bezpośrednio po niej zaproszenie (tzw. *prompt*) powłoki.

4.1.4 Ocena/wnioski/komentarze dotyczące wykonania danego zadania.

### 2.1) Filtry

Nazwa polecenia	cut
Opis działania polecenia	Składnia: cut OPCJA... [PLIK]... Wypisywanie wybranych części linii z każdego PLIKU na standardowe wyjście.
Podstawowe parametry wywołania	-d, --delimiter separator pól -f, --fields pola do wyświetlenia -c, --characters znaki do wyświetlania -b, --complement wyświetla wszystkie pola które nie zostały wskazane (tzw. dopełnienie)
Przykłady użycia (polecenie i parametry)	
Objaśnienie wykonanej funkcji	

Nazwa polecenia	grep
Opis działania polecenia	konfigurowalny filtr - przepuszcza z <b>stdin</b> do <b>stdout</b> linie które pasują do zadanego wzorca ( <i>globalnego / wyrażenia regularnego / wypisywania na ekran</i> )
Podstawowe parametry wywołania	-i, --ignore-case nie zważaj na wielkość liter -v, --invert-match przepuść tylko linie które nie pasują do wzorca (dopełnienie)

	-e, --regexp=PATTERN zastosuj wyrażenie regularne do filtrowania
Przykłady użycia (polecenie i parametry)	
Objaśnienie wykonanej funkcji	

Nazwa polecenia	head
Opis działania polecenia	Służy do pokazywania pierwszych linii bądź znaków z podanych plików, dla każdego oddzielnie. Domyślnie wyświetla dziesięć pierwszych linii, ale zachowanie to może zostać zmienione odpowiednimi przełącznikami.
Podstawowe parametry wywołania	-c num – wypisuje num pierwszych bajtów w pliku; można używać przedrostków: b jako 512 bajtów k jako kilobajt m jako megabajt -n num – wypisuje num pierwszych linii w pliku
Przykłady użycia (polecenie i parametry)	
Objaśnienie wykonanej funkcji	

Nazwa polecenia	sort
Opis działania polecenia	Polecenie sort układa wiersze pliku w określonym porządku zgodnie z wartościami pewnych pól zawartymi w każdym pliku. Pola takie nazywamy kluczami sortowania np. kluczem sortowania w pliku zawierającym dane każdego obywatela może być numer pesel lub nazwisko.
Podstawowe parametry wywołania	-t włącza znak rozdzielający inny niż spacja lub tabulator -d sortowanie w kolejności książki telefonicznej -f sortowanie nie rozróżniające małych i wielkich liter -i sortowanie ignorujące znaki spoza zakresu ASCII -n sortowanie najpierw w kolejności liczbowej, a potem alfabetycznej -r sortowanie odwracające kolejność danych wynikowych
Przykłady użycia (polecenie i parametry)	sort plik
Objaśnienie wykonanej funkcji	Sortuje alfabetycznie plik plik z katalogu bieżącego

Nazwa polecenia	tail
Opis działania polecenia	wyświetla koniec pliku (strumienia) - domyślnie 10 linii
Podstawowe parametry wywołania	-n, --lines=K      czytaj K ostatnich linii; jeśli K zaczyna się od znaku +, czytaj wszystkie linie pliku oprócz K pierwszych -f, --follow      nie przerywaj czytania, wyświetlaj dane dopisywane do pliku w miarę jak się pojawiają
Przykłady użycia (polecenie i parametry)	tail -25 /var/log/messages
Objaśnienie wykonanej funkcji	Wyświetla ostatnich 25 linii pliku /var/log/messages

Nazwa polecenia	tee
-----------------	-----

Opis działania polecenia	Pobiera dane ze standardowego wejścia i wypisuje je na standardowe wyjście oraz do wskazanych plików (o ile jakieś zostały podane).
Podstawowe parametry wywołania	-a lub --append - jeśli plik już istnieje, nie jest nadpisywany, a zawartość wejścia jest do niego dopisywana; -i lub --ignore-interrupts - ignoruje sygnały przerwania (SIGINT);
Przykłady użycia (polecenie i parametry)	
Objaśnienie wykonanej funkcji	

Nazwa polecenia	tr
Opis działania polecenia	zmienia lub usuwa znaki ze standardowego strumienia wejścia (stdin)
Podstawowe parametry wywołania	<p>-s, --squeeze zamiast tłumaczyć, usuń powtórzenia</p> <p>-d, --delete zamiast tłumaczyć, usuń znaki</p> <p>--complement zamiast wykonywać operacje dla wskazanego zbioru znaków, wykonuje ją dla wszystkiego co nie mieści się we wskazanym zbiorze znaków (tzw. dopełnienie)</p> <p>Zamiast samego znaku możliwe jest również podanie odpowiedniej sekwencji specjalnej:</p> <p>\a – dźwięk dzwonka (numer 7 w ASCII);</p> <p>\b – cofnięcie się o jeden znak w tył (backspace, 8 w ASCII);</p> <p>\f – znak końca strony (form feed, 12 w ASCII);</p> <p>\n – znak nowej linii (10 w ASCII);</p> <p>\r – znak powrotu karetki (13 w ASCII);</p> <p>\t – znak tabulacji (9 w ASCII);</p> <p>\v – znak pionowej tabulacji (11 w ASCII).</p> <p>Ponadto cały ciąg może być sekwencją specjalną:</p> <p>ZNAK1-ZNAK2 – wszystkie znaki od pierwszego do drugiego w porządku alfabetycznym;</p> <p>[:alnum:] – wszystkie litery i cyfry;</p> <p>[:alpha:] – wszystkie litery;</p> <p>[:blank:] – wszystkie znaki białe;</p> <p>[:cntrl:] – wszystkie znaki kontrolne;</p> <p>[:digit:] – wszystkie cyfry;</p> <p>[:graph:] – wszystkie znaki drukowalne, oprócz spacji;</p> <p>[:lower:] – wszystkie małe litery;</p> <p>[:print:] – wszystkie znaki drukowalne, w tym spacja;</p> <p>[:punct:] – wszystkie znaki interpunkcyjne;</p> <p>[:space:] – wszystkie znaki białe;</p> <p>[:upper:] – wszystkie wielkie litery.</p> <p>Niestety, sekwencje te mają problemy ze znakami spoza zestawu ASCII, w tym z polskimi literami.</p>
Przykłady użycia (polecenie i parametry)	tr -d "\n " < /var/log/messages > mes.kopia
Objaśnienie wykonanej funkcji	Polecenie wykonuje kopie pliku /var/log/messages i zapisuje ją do pliku mes.kopia w katalogu bieżącym użytkownika.

Nazwa polecenia	<code>uniq</code>
Opis działania polecenia	Usuwa i/lub zlicza powtarzające się/unikalne linie, działa na posortowanych danych
Podstawowe parametry wywołania	<code>-c, --count</code> policz powtarzające się linie
Przykłady użycia (polecenie i parametry)	<code>uniq -d -i</code>
Objaśnienie wykonanej funkcji	Podaje jakie linie są takie same

Nazwa polecenia	<code>wc</code>
Opis działania polecenia	liczy znaków/słów/linii
Podstawowe parametry wywołania	<code>-c</code> lub <code>--bytes</code> – liczba bajtów, <code>-l</code> lub <code>--lines</code> – liczba linii, <code>-L</code> lub <code>--max-line-length</code> – długość najdłuższej linii, <code>-m</code> lub <code>--chars</code> – liczba znaków, <code>-w</code> lub <code>--words</code> – liczba słów.
Przykłady użycia (polecenie i parametry)	<code>wc -w /var/log/messages</code>
Objaśnienie wykonanej funkcji	Pokazuje ilość słów w pliku <code>/var/log/messages</code>

Zad. 3. Posługując się poleceniem `tr` wykonaj kopię pliku `/var/log/dmesg` lub `/var/log/messages` pozbawioną znaków nowej linii.

```
[root@centos75 ~]# tr -d "\n" < /var/log/messages > mes.kopia
```

Zad. 4. Za pomocą polecenia `cat` utwórz kopię pliku `/var/log/boot.log` z ponumerowanymi wierszami, lecz bez pustych linii.

```
[root@centos75 ~]# grep [[:blank:]] /var/log/boot.log | cat -n > log.boot
```

Zad. 5. Wyświetl 25 ostatnich linii pliku `/var/log/messages` i zlicz znajdujące się tam słowa.

```
[root@centos75 ~]# tail -25 /var/log/messages
```

```
[root@centos75 ~]# tail -25 /var/log/messages | wc -w
```

Zad. 6. Za pomocą pojedynczego polecenia wyświetl komunikat (bez cudzysłowów):

„Plik `/etc/passwd` zawiera xxx linii i yyy znaków.”

Wartości xxx i yyy winny być obliczone przez odpowiednie polecenia powłoki.

```
[root@centos75 ~]# echo "Plik /etc/passwd `wc -l < /etc/passwd` linii i  
`wc -m < /etc/passwd` znakow."
```

Zad. 7. Za pomocą pojedynczego polecenia wyświetl na konsoli plik `/etc/system-release` i równocześnie utwórz jego kopię `/etc/system-release.copy`.

```
echo "`cat /etc/system-release` `cat /etc/system-release > /etc/system-release.copy`"
```

## 2.2) Potoki

Zad. 1. Polega na połączeniu dwóch strumieni danych. Dane są pobierane z pierwszego z nich i przekazywane do drugiego. Zazwyczaj strumieniami tymi są standardowe wyjście i standardowe wejście, by go użyć, należy użyć znaku kreski pionowej „|”.



Zad. 2. Za pomocą potoku wyświetl wykaz plików (bez katalogów) znajdujących się w bieżącym katalogu.

```
[root@centos75 ~]# ls -a -p | grep -v /
```

Zad. 3. Zmodyfikuj poprzednie zadanie tak, aby wykaz był posortowany według rosnącej objętości plików. Lista ma zawierać nazwę pliku i jego objętość.

```
[root@centos75 ~]# ls -p -a -s | grep -v /|sort
```

Zad. 4. Za pomocą potoku oblicz łączną liczbę katalogów znajdujących się w bieżącym katalogu i jego podkatalogach.

```
[root@centos75 ~]# find | wc -l
```

Zad. 5. Utwórz potok, który wyświetli uporządkowaną alfabetycznie listę ostatnio wydanych różnych poleceń powłoki.

```
[root@centos75 ~]# history | cut -c 6- | sort
```

Zad. 6. Zmodyfikuj poprzednie zadanie tak, aby wyświetlić tylko 10 alfabetycznie pierwszych poleceń powłoki.

```
[root@centos75 ~]# history | head | sort
```

## 5. Wnioski z przeprowadzonych prac

(podsumowanie celu ćwiczenia i osiągniętych wyników, wnioski dotyczące zastosowanych środków programowych i uzyskanych wyników, samoocena stopnia osiągnięcia celu ćwiczenia)

## 6. Inne uwagi