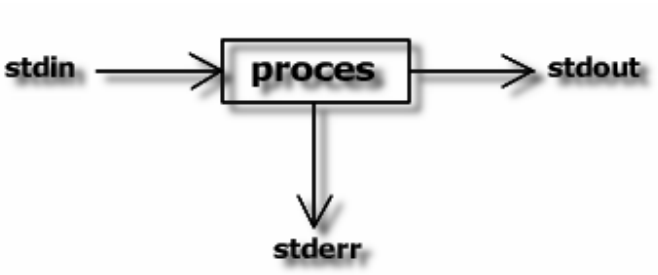
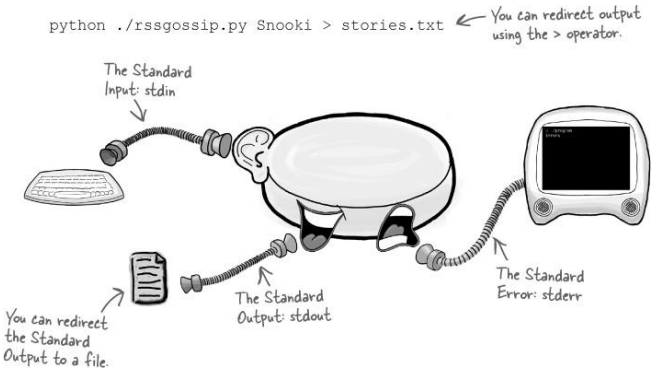
Przekierowanie strumienia danych.





Powłoki w systemach uniksowych intensywnie wykorzystują strumienie: wejścia, wyjścia i błędów. Strumienie można przekierować z urządzenia, które je obsługuje na inne lub też do pliku, np. można przekierować komunikaty o błędach z monitora do pliku. Strumienie mają swój unikalne deskryptory:

a) stdout - standardowe wyjście (deskryptor 1) - przeważnie jest to monitor komputera,

b) stderr - standardowe wyjście błędów (deskryptor 2) - przeważnie jest to monitor komputera,

c) stdin - standardowe wejście (deskryptor 0) - przeważnie jest to klawiatura komputera.

Po wykonaniu poniższych czynności wykonaj notatkę w zeszycie.

Przekierowania zazwyczaj wykorzystywane są do odczytania danych z jakiegoś urządzenia lub pliku albo do wysłania danych na urządzenie lub plik. Do przekierowań danych służą specjalne znaki:

< dane na wejście

> dane z wyjścia

Najprostszym przykładem przekierowania może być wyświetlenie zawartości jakiegoś pliku poleceniem:

cat < plik.txt

Aby utworzyć nowy plik i wpisać do niego dowolny tekst użyj przekierowania danych z wyjścia:

cat > plik.txt

Polecenie to utworzy plik plik.txt i umieści w nim tekst wpisany z klawiatury. Jeśli będziemy chcieli ponownie dodać jakiś tekst do tego pliku należy użyć polecenia:

cat >> plik.txt

Spowoduje to dodanie następnych wierszy z informacją do pliku (bez kasowania poprzedniej zawartości).

Innym przykładem przekierowań może być zapisanie do pliku zawartości jakiegoś katalogu:

ls -la > plik.txt

grep – służy do filtracji strumieni, wyświetla linie pasujące (lub nie) do określonego wzorca. "grep" jest rozbudowaną komendą, lecz zwykle administratorzy używają kilka jego podstawowych właściwości.

Strumień jest odczytywany linia po linii, na wyjście przedostają się tylko te linie, które są zgodne z podanym wzorcem. Tego polecenia używa się najczęściej z pipeline | :

cat /etc/passwd | grep user – czyta plik z /etc/passwd i wypisuje na ekranie tylko linię z łańcuchem user

Różnica między | a > jest następująca:

| służy do łączenia wyjścia jednego procesu z wejściem drugiego

> służy do określania pliku, który będzie pełnił rolę wyjścia dla procesu

cat /etc/passwd | grep user

oznacza odczytanie pliku /etc/passwd przez program cat i przekierowanie wyjścia na wejście grep'a wywołanego z parametrem user. Innymi słowy: uruchomione dwa procesy (cat i grep) połączone ze sobą.

cat /etc/passwd > grep user

nie ma zbyt dużego sensu. Oznacza ono uruchomienie cata, którego wyjście zostanie przekierowane do pliku grep (fraza user będzie poddana próbie interpretacji jako parametr albo nazwa pliku).

Zauważmy, że **cat /etc/passwd > grep** oznaczałoby wylistowanie całej zawartości pliku /etc/passwd do pliku tekstowego o nazwie grep!

Uproszczona składnia: grep [-v] WZORZEC [PLIK(I)] Najważniejsze operatory wyrażeń regularnych:

Znak	Opis
.	Dopasuj dowolny znak
$	Dopasuj poprzedzające wyrażenie do końca wiersza
^	Dopasuj występujące po operatorze wyrażenie do początku wiersza
*	Dopasuj zero lub więcej wystąpień znaku poprzedzającego operator
\	Oznacza pominięcie specjalnego znaczenia znaku np.: *
[]	Dopasuj dowolny znak ujęty w nawiasy. np.: [abc]
[-]	Dopasuj dowolny znak z przedziału. np.: [0-9] – wszystkie cyfry; [a-z] – wszystkie małe litery; [0-9a-zA-Z] – wszystkie litery i cyfry

[^]	Dopasuj znak, który nie znajduje się w nawiasach.

Przykłady:

grep 'Ala' plik - znajduje wyraz Ala

grep 'A.a' plik - znajduje wyrazy takie jak Ala, Aga, Ara, A+a i inne

grep 'A[lg]a' plik - znajduje TYLKO wyrazy Ala i Aga grep '^Ala' plik

Linux wszystko traktuje jako plik, niezależnie od tego czy to jest plik zwykły, katalog, urządzenie blokowe (klawiatura, ekran) itd. Nie inaczej jest ze strumieniami, BASH identyfikuje je za pomocą przyporządkowanych im liczb całkowitych (od 0 do 2) tak zwanych deskryptorów plików.

0 to plik, z którego proces pobiera dane stdin

1 to plik, do którego proces pisze wyniki swojego działania stdout

2 to plik, do którego trafiają komunikaty o błędach stderr

2> – przekierowanie standardowego wyjścia dla błędów do pliku. Poza standardowym wyjściem i wejściem, program ma jeszcze standardowe wyjście dla błędów. Je też możemy przekierowywać, na przykład w inne miejsce niż wyjście zwykłe. Działa w "bash", nie w "tcsh".

find -name "plik.*" >znalezione.log 2>bledy.log – pliki znalezione przez 'find' wylądują w znalezione.log, komunikaty o błędach nie zaciemnią nam wyniku i wpiszą się do innego pliku bledy.log

cp -r /dane /backup 2>error.log – jeśli podczas kopiowania całego katalogu wystąpią błędy, wszystkie komunikaty zostaną wpisane do error.log

(ls > plik.log) >& plik.err – w 'tcsh' nie można przekierować samego wyjścia dla błędów, stąd konieczność takiej konstrukcji.

&> lub >& – przekierowanie obu wyjść do pliku.

ls >& plik.log – standardowe wyjście i standardowe wyjście dla błędów są przekierowane do plik.log

ls >plik.log 2>&1 – to samo, ale działa tylko w 'bash'. Przekierowanie wyjścia, a potem skopiowanie go na wyjście dla błędów.

ls &> plik.log – to samo co >& ale w notacji bardziej właściwej dla 'bash'.

rm -rf /tmp 2> /dev/null - spowoduje usunięcie wszystkich plików z katalogu /tmp, do których dany użytkownik ma prawa pisania ("w"), a komunikaty o błędach zostaną przekierowane do /dev/null, czyli nie istniejącego zasobu

bash < /tmp/in >>/tmp/wynik 2>/dev/null

Polecenie to uruchomi kopię powłoki bash, wczyta z pliku /tmp/in wszystkie wpisane tam komendy, wykona je, wynik pracy dołączy do pliku /tmp/wynik, a komunikaty o wszystkich błędach, które potencjalnie wystąpią, zostaną zignorowane. (śmietnika)

head, tail

Polecenie head i tail pozwalają na wyświetlanie części pliku: odpowiednio początku i końca. Np.:

head -n 10 plik wyświetli pierwszych 10 linii pliku.

Polecenie tail może spełnia szczególną funkcję po wywołaniu z parametrem "-f". Wyświetla koniec pliku i oczekuje na nowe dane. Może, więc służyć jako „monitor" pliku modyfikowanego przez inną aplikację (pracującą w tle lub na innej konsoli). Np.:

wget -t0 -rl5 -oout.txt -Pwp http://www.zsl.gda.pl

tail –f out.txt

more, less

Polecenie more i less pozwalają na łatwiejsze przeglądanie strumienia wyjściowego.
W przypadku dużej ilości danych konsola systemu zostaje „przewinięta" i część danych zostaje utraconych. Istnieje możliwość cofnięcia tekstu na konsoli (shift+pgup lub suwak w xterm) jednak bufor danych jest także ograniczony. Aby móc swobodnie czytać dane zwracane przez program wywołujemy polecenie: komenda | more

Po każdym zapełnieniu ekranu more zatrzyma wyświetlanie danych i będzie oczekiwał na naciśnięcie dowolnego klawisza. Polecenie less jest wygodniejsze, gdyż pozwala na swobodne przewijanie danych. Pracę z danymi wyświetlonymi przez less możemy zakończyć naciskając klawisz 'q'.

sort, uniq

Komenda sort służy do sortowania (domyślnie: alfabetycznego) linijek tekstu stanowiących dane wejściowe. Gdy się ją wywoła z argumentami będącymi nazwami plików, danymi do sortowania będzie zawartość tychże; w przypadku wywołania bez argumentów (nie będących opcjami, za pomocą, których można zadać bardziej złożone kryteria sortowania), komenda sort oczekuje, że dane do przetworzenia pojawią się w standardowym strumieniu wejściowym. W obu tych przypadkach, wynik sortowania pojawi się na stdout. Przykład: cat /etc/passwd | sort zwróci posortowaną listę użytkowników systemu.

Uzupełnieniem komendy sort jest komenda uniqe. Powoduje pominięcie wierszy powtarzających się. Np.: cat /etc/passwd | sort | uniq

Zgłoszenie 1

W celu powtórzenia wykonaj w domu następujące zadania

1. Przekieruj strumienie wyjściowe kilku komend do pliku, a następnie przejrzyj ich zawartość.
2. Uzyskaj w pliku dane informacje na temat błędów popełnionych w wywoływanych komendach.
3. Dopisz do pliku już uzyskanego stout i sdterr na dwa sposoby.
4. Wykonaj przekierowanie wszystkich 3 strumieni.
5. Wygeneruj listę wszystkich plików w systemie, posortuj a następnie usuń duplikaty, zapisz to wszystko do pliku.
6. Z listy, którą otrzymałeś w poprzednim ćwiczeniu wygeneruj listę plików nagłówkowych (*.h) i zapisz do pliku.
7. Sprawdź parametry polecenia sort – czy pozwala ono na sortowanie według innych zasad niż „alfabetycznie”?