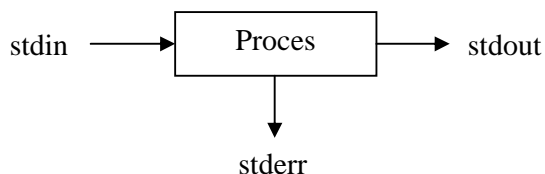


Przekierowanie strumienia danych

Każdy uruchomiony w Linuksie proces pobiera z pewnego miejsca dane, a w inne wysyła wyniki swojego działania i komunikaty o błędach. Dane są przesyłane między urządzeniami w postaci **strumieni**. Strumienie danych przypisane każdemu procesowi są pokazane na schemacie:



stdin (*standard input*) – standardowe wejście, z którego proces pobiera dane (domyślnie jest to klawiatura),

stdout (*standard output*) – standardowe wyjście, do którego wysyłany jest wynik działania procesu (domyślnie jest to ekran),

stderr (*standard error*) – standardowe wyjście błędów, gdzie trafiają wszystkie komunikaty o błędach (domyślnie ekran).

Standardowe wejście i wyjście (strumienie danych) można przekierować. Do tego celu przygotowano trzy operatory:

- znak „<” umożliwia przekierowanie zawartości pliku do standardowego wyjścia, np. `more < plik`,
- znak „>” umożliwia przekierowanie strumienia danych ze standardowego wyjścia do pliku; jeżeli plik istnieje, to jego poprzednia zawartość zostaje usunięta, np. `ls > plik`,
- znaki „>>” umożliwiają przekierowanie strumienia danych ze standardowego wyjścia do pliku; jeżeli plik istnieje, to nowe dane zostają dopisane na końcu pliku.

Przełączanie standardowego wejścia

Jako standardowe wejście można zamiast klawiatury użyć pliku:

```
< plik
```

Ćwiczenie 1 – przekierowanie strumienia danych z pliku

1. Utwórz plik **lista** o następującej zawartości:
Matematyka
Historia
Język polski
Biologia
2. Następnie użyj polecenia `sort`, dla którego standardowym wejściem będzie plik **lista**:
`sort < lista`. Wynikiem będzie wyświetlenie na ekranie posortowanej zawartości pliku **lista**.

Przełączanie standardowego wyjścia

Wynik dowolnego polecenia można wysłać do pliku, a nie na ekran. Do tego celu używa się operatora:

```
> plik
```

Przykład:

```
ls -la /usr/bin > wynik
```

Rezultat działania powyższego polecenia trafi do pliku `wynik`, jeśli wcześniej nie istniał plik o takiej samej nazwie, to zostanie utworzony, jeśli istniał, cała jego poprzednia zawartość zostanie nadpisana.

Jeśli użytkownik chce, aby dane wyjściowe dopisywane były na końcu pliku, bez wymazywania jego wcześniejszej zawartości, to powinien stosować operator:

```
>> plik
```

Przykład:

```
free -m >> wynik
```

Wynik tego polecenia (pokazanie wykorzystania pamięci RAM i swap) zostanie dopisany na końcu pliku `wynik`, nie naruszając jego wcześniejszej zawartości.

Ćwiczenie 2 – przekierowanie strumienia danych do pliku

1. Zaloguj się na konto użytkownika `root`.
2. Wpisz polecenie: `ls /etc > ~/lista_plikow.txt`
Jeżeli plik **lista_plikow.txt** istniał w katalogu domowym, to zostanie usunięty i utworzony nowy plik o tej samej nazwie.
3. Wpisz polecenie: `date >> ~/lista_plikow.txt`
Na końcu pliku **lista_plikow.txt** zostanie dopisana informacja o czasie wywołania polecenia.
4. Sprawdź zawartość pliku **lista_plikow.txt** poleceniem: `cat ~/lista_plikow.txt`

Przełączanie standardowego wyjścia błędów

Do pliku można też kierować strumień diagnostyczny:

```
2> plik
```

Przykład:

```
ls -y 2> ~/error
```

W powyższym skrypcie polecenie `ls` jest użyte z błędną opcją `-y`; komunikat o błędzie trafi do pliku `error`.