

04.01.2010

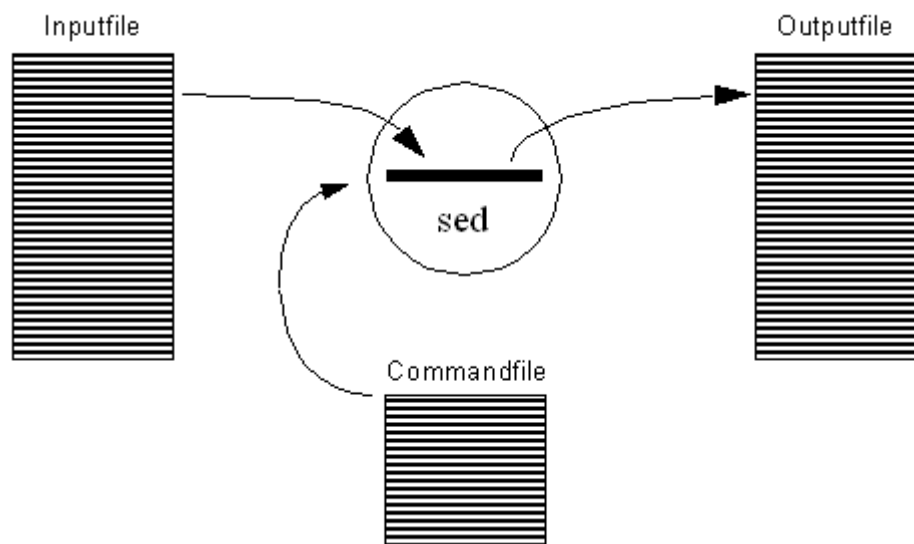
SED - Stream EDitor

edytor strumieniowy

Bogumił Konopka
W-11/I-21
Politechnika Wrocławska

Jak działa sed?

- sed jest filtrem – pobiera dane w postaci wierszy tekstu ze standardowego wejścia
- przetwarza je zgodnie z poleceniami podanymi w wierszu poleceń lub zapisanymi w pliku
- przesyła wyniki operacji na standardowe wyjście.



Zalety

- Dostępność – sed jest dostępny we wszystkich aktualnych dystrybucjach Linuksa
- Prostota – jest narzędziem prostym przy jednocześnie szerokim wachlarzu zastosowań
- Brak ograniczeń rozmiarami plików ze względu na przetwarzanie wiersz po wierszu

Przykład 1!

Składnia

- `sed [-n] [-g] [-e skrypt] [-f splik] [plik]`
 - `n` hamuje normalne wyjście
 - `g` powoduje, że `sed` zachowuje się tak, jakby każda z komend skryptu miała przyrostek `g`
 - `e` załącza pojedynczą komendę edycyjną z następującego po niej argumentu; jeśli komend jest więcej, to są one wykonywane w kolejności pojawienia się
 - `-f` powoduje, że komendy są pobierane z pliku “`splik`”; jeśli jest ich kilka, to są one wykonywane w kolejności pojawienia się;

Polecenia

- Komendy:
 - “d” – delete – usuwanie
 - “p” – print – drukowanie
 - “s” – substitute – zamiana
- Adresowanie:
 - Numer wiersza
 - Zakres wierszy
 - Wyrażenia regularne

Proste polecenia - przykłady

- `sed -ne 'p' input` // wydrukuje cały plik
- `sed -e 'd' input` // nie wydrukuje (skasuje) niczego
- `sed -e '1d' input` // nie wydrukuje tylko pierwszej lini
- `sed -e '1,10d' input` // nie wydrukuje lini od 1-10
- `sed -e 'p' input` // wydrukuje każdą linię dwa razy plik

Przykład 2!

Adresowanie z wykorzystaniem wyrażeń regularnych

- Wyrażenia regularne – wzorce opisujące łańcuch symboli, algorytmy w efektywny sposób określają, czy podany ciąg znaków pasuje do wzorca lub wyszukują w tekście wystąpienia wzorca.

Wyrażenia regularne

Znak	Znaczenie
<code>^</code>	Początek wiersza
<code>\$</code>	Koniec wiersz
<code>.</code>	Dowolny znak
<code>*</code>	0 lub więcej powtórzeń poprzedniego znaku
<code>[]</code>	Znak pomiędzy “[” i “]”

Wyr. regularne	Znaczenie
<code>/./</code>	Wiersz zawierający co najmniej jeden znak
<code>/../</code>	Wiersz zawierający co najmniej dwa znaki
<code>/[abc]/</code>	Wiersz zawierający a, b lub c
<code>/[abc]/</code>	Wiersz zaczynający się od a,b lub c
<code>/^#/</code>	Wiersz zaczynający się od #
<code>/}^/</code>	Wiersz kończący się “}”

Adresowanie z wykorzystaniem wyrażień regularnych - przykłady

- `sed -ne /DSSP/p input //` wydrukuje linie z DSSP
- `sed -ne /[sS]bjct/p input //`wydrukuje linie z sbjct lub Sbjct
- `sed -ne '/^\s.*[1-9]:/p' input //`wydrukuje linie zaczynające się spacją oraz zawierające liczbę po której następuje “:”

Przykład 3

Adresowanie od - do

- `sed -n -e '/POCZATEK/,/KONIEC/p' input`
- Polecenie wydrukuje cały tekst od znalezienia wzorca POCZATEK, aż do natrafienia na wzorzec KONIEC
- `sed -ne '/<PRE>/,/<VPRE>/p' input`

Przykład 4

Polecenie zamiany - "s"

- Polecenie podstawienia pozwala wymienić określony ciąg znaków innym ciągiem znaków

`s/ / /`

- `sed -e 's/PDB/database/' input`
- `sed -e '1,50s/MOLECULE/PROTEIN/' input`
//polecenie zmieni w liniach 1-50 wyraz MOLECULE na PROTEIN
- `sed -e '1,50!s/MOLECULE/PROTEIN/' input`
//polecenie zmieni wyraz MOLECULE na PROTEIN wszędzie poza l1 - 50

W instrukcji zamiany również możemy korzystać z wyrażeń regularnych

- `sed -e '1,50s/[a-z0-9]\{4\}/protein/' input`
- `sed -e '1,50s/ [a-z0-9]\{4\}/protein/' input`

Zaawansowane podstawianie 1

- Gdy chcemy dodać coś do linii:
 - & - zmienna przechowująca łańcuch pasujący do wzorca
- `sed -e '1,50!s/MOLECULE/& PROTEIN/' input`
// komenda zastąpi wystąpienia wyrazu “MOLECULE” przez “MOLECULE PROTEIN”

Przykład 6

Zaawansowane podstawianie 2

- Definiowanie regionów – umożliwia dowolne odwoływanie się do poszczególnych z nich
- Wyrażenie umieszczane jest w nawiasach poprzedzonych ukośnikiem “\(\wyr\)”
- Odwołanie odbywa się poprzez liczbę porządkową regionu poprzedzoną ukośnikiem np. “\1”

```
sed -n -e 's/.*[0-9]:[\t ]*\([a-z0-9]*-[A-Z ]\).*\1/p' input
```

Przykład 7

Inne przykłady

- Pre-processing bazy danych
- Przeformatowanie pliku wejściowego
- Usuwanie zbędnych linii

Źródła i dodatkowe informacje

- <http://www.gentoo.org/doc/pl/articles/l-sed1.xml>
- <http://www.gentoo.org/doc/pl/articles/l-sed2.xml>
- <http://www.gentoo.org/doc/pl/articles/l-sed3.xml>
- <http://www.grymoire.com/Unix/Sed.html#uh-nl>
- <http://www.grymoire.com/Unix/Regular.html>
- http://www.lug.or.kr/files/cheat_sheet/regular-expressions-cheat-sheet-v2.png
- <http://www.gnu.org/software/sed/manual/sed.html>