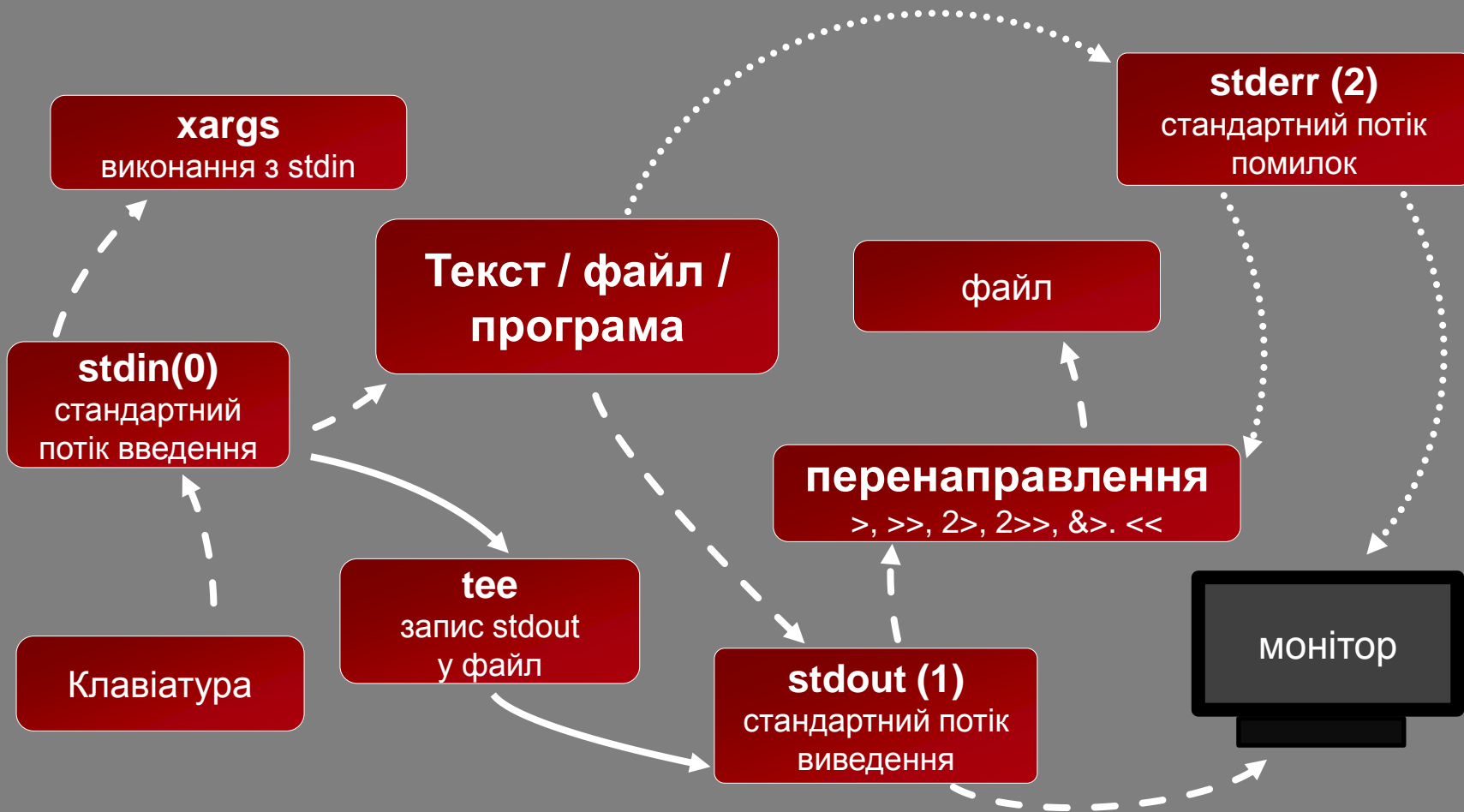


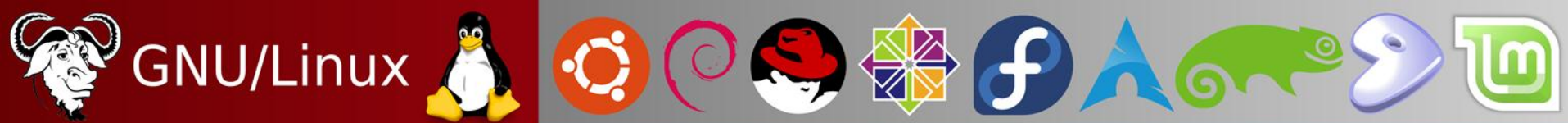
# Стандартні потоки Linux

## Використання потоків та перенаправлень

- Стандартний потік введення
- Стандартний потік виведення
- Стандартний потік помилок
- Символи перенаправлення
- tee
- xargs

Потоки даних





## Використання потоків та перенаправлень

- Стандартним потоком введення (stdin) є клавіатура. У більшості випадків, потік вхідних даних вводиться саме через клавіатуру.
- Стандартний потік виведення (stdout) - це потік, в який програми виводять результати роботи. Зазвичай ці дані відображаються на екрані монітора.
- Стандартний потік помилок (stderr) - потік, який використовується для виведення повідомлень про помилки і діагностики. Цей потік незалежний від стандартного потоку виводу і може перенаправлятися окремо.

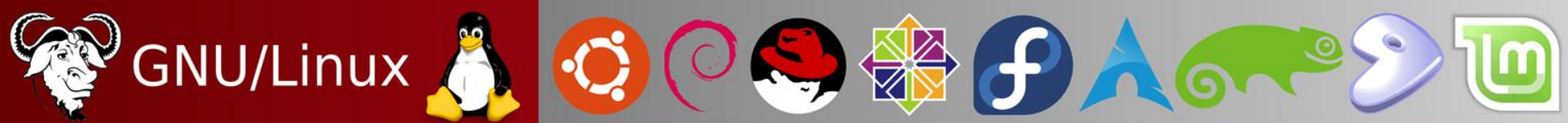
# Символи перенаправлення даних

<b>&gt;</b>	Створює новий файл, який містить дані стандартного потоку виводу. Якщо вказаний файл існує, то він перезаписується.
<b>&gt;&gt;</b>	Створює новий файл, який містить дані стандартного потоку виводу. Якщо вказаний файл існує, то він дописується.
<b>2&gt;</b>	Створює новий файл, який містить дані стандартного потоку помилок. Якщо вказаний файл існує, то він перезаписується.
<b>2&gt;&gt;</b>	Додає інформацію зі стандартного потоку виводу в існуючий файл. Якщо вказаний файл не існує, то він створюється.
<b>&amp;&gt;</b>	Створює новий файл, який містить дані стандартного потоку виводу і потоку помилок. Якщо вказаний файл існує, то він перезаписується.



# Символи перенаправлення даних

<b>2&gt;&amp;1</b>	Виводить стандартні потоки виведення і помилок в один файл.
<b>&lt;</b>	Перенаправляє вміст файлу як стандартний потік введення.
<b>&lt;&lt;</b>	"Heredoc-синтаксис" - спеціальна форма введення для формування вхідного потоку програм.
<b> </b>	Перенаправляє стандартний потік виведення однієї команди в стандартний потік введення іншої команди.



## Опції команди xargs

<b>-0 (нуль)</b>	Замінює стандартний роздільник (перенесення рядка на нуль-символ (\x0), який і означає кінець збереженого рядка. Використовується для вирішення проблеми наявності прогалів в імені переданого параметра.
<b>-n число</b>	Вказує максимальну кількість аргументів, які можна передати команді за одну підстановку.
<b>-p</b>	Запитує підтвердження перед кожним викликом команди.
<b>-t</b>	Перед виконанням вивести команду з побудованим списком аргументів на стандартний потік помилок.
<b>-P число</b>	Вказує максимальну кількість процесів, які будуть виконуватися одночасно тобто кількість запускених примірників команди.

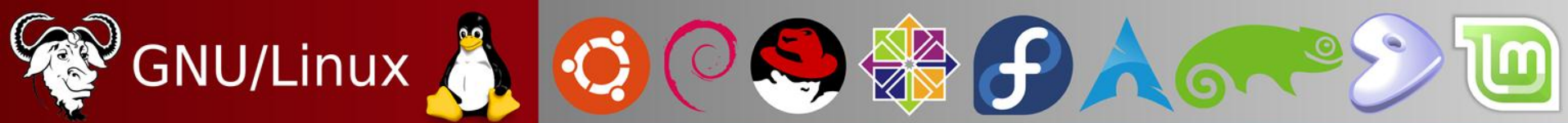
# Регулярні вирази





## Регулярні вирази

- Будь-який текст, таблиці з даними, списки файлів містять в собі певний набір символів. Часто виникає завдання пошуку слів або виразів, що належать до одного типу, але з можливими варіаціями в написанні, такі як дати, імена файлів, e-mail адреси. З іншого боку, є завдання по знаходженню цілком певних слів, які можуть мати різне написання, або пошук, що виключає окремі символи або класи символів. Для цих цілей були створені певні системи, засновані на описі тексту за допомогою шаблонів. До таких систем відносяться і регулярні вирази



## Шаблон пошуку

- Будь-який пошук заснований на пошуку за певним зразком - шаблоном або маскою пошуку. Шаблон пошуку задається за допомогою спеціальних символів: символи-джокери або метасимволи - символи, які використовуються для заміни інших символів або їх послідовностей, приводячи таким чином до символічних шаблонів. Основні такі символи відомі і часто використовуються при пошуку файлів на диску або інформації в пошукових системах.

## Команда grep

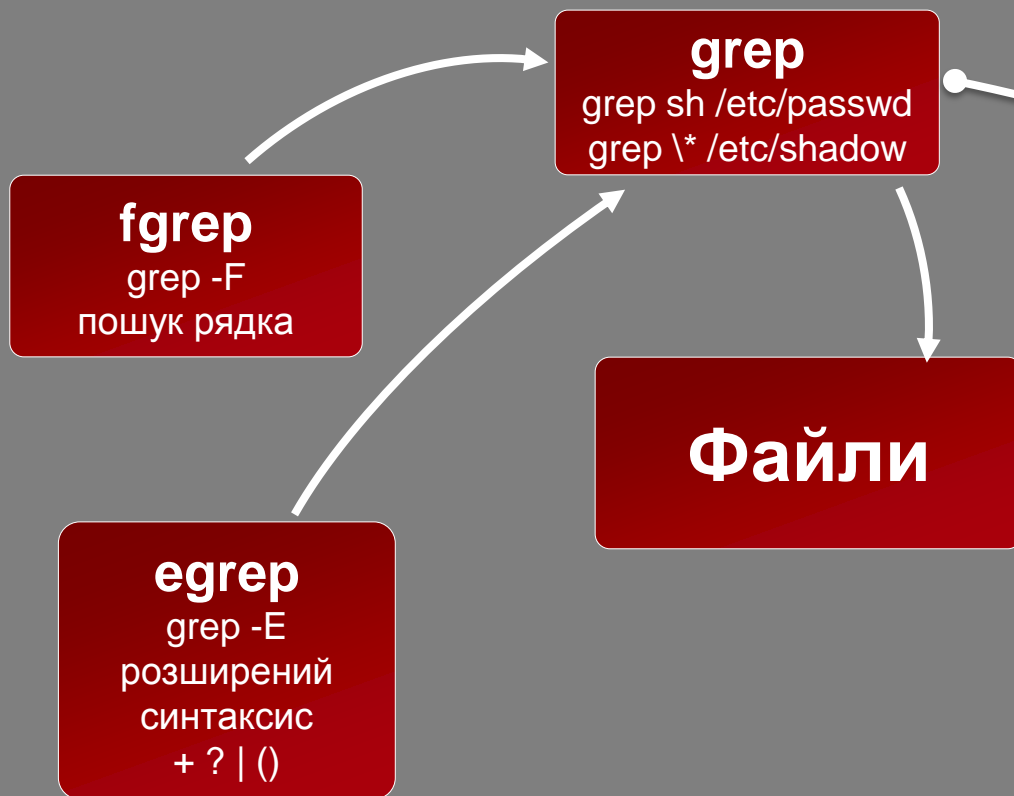
- **Регулярні вирази** (англ. Regular expressions, скор. RegExp, RegEx) – система синтаксичного розбору текстових фрагментів за формалізованим шаблоном, заснована на системі запису зразків для пошуку. Зразок (англ. Pattern) задає правило пошуку, також іноді називається «шаблоном», «маскою».
- **grep** – утиліта командного рядка, яка знаходить рядки, що відповідають заданому регулярному виразу, і виводить їх. Назва являє собою акронім англійської фрази «search Globally for lines matching the Regular Expression, and Print them» – «шукати всюди рядки, що відповідають регулярному виразу, і виводити їх».

# Використання регулярних виразів

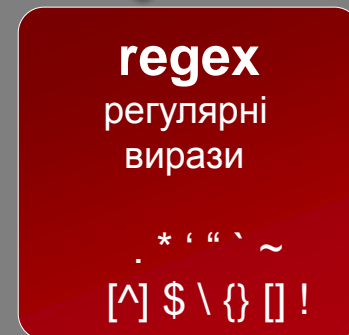
- regex
- grep
- egrep
- fgrep



## Пошук за допомогою grep



## Регулярні вирази



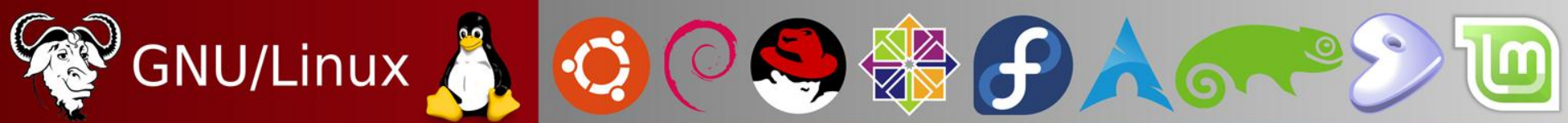


## Регулярні вирази

'	Змінні і команди всередині одинарних лапок не обробляються.
"	Змінні всередині подвійних лапок обробляються, а команди - ні.
`	Усередині зворотніх лапок обробляються і змінні, і команди.
\	Екранування спеціальних символів.
~	Символ тильди відповідає домашньому каталогу користувача.

# Регулярні вирази

.	Будь-який символ
[ ]	Відповідає будь-якому символу, що знаходиться всередині дужок
[^]	Відповідає будь-якому символу, крім того, що знаходиться всередині дужок
*	Відповідає повторенню попереднього символу нулю або більше раз
{m,n}	Визначення допустимої кількості повторень виразу
^	Символ ^ позначає початок рядка
\$	Символ \$ відповідає кінцю рядка
()	Групування виразів



# Символьні класи

<b>[0-9]</b>	Цифри
<b>[A-Z]</b>	Великі літери
<b>[a-z]</b>	Маленькі літери



## Опції команди grep

<b>-E</b>	Розширений синтаксис (Extended)
<b>-F</b>	Зіставлення з фіксованими рядками (Fixed)
<b>-I</b>	Відключення чутливості до регістру
<b>-n</b>	Виводити номери рядків, в яких знайдено збіг
<b>-R</b>	Рекурсивний пошук у вказаному каталозі
<b>-v</b>	Інверсія результатів пошуку

## Команда egrep

- Відмінність від grep полягає в можливості використовувати розширені регулярні вирази
- Розширені регулярні вирази або ERE (Extended Regular Expressions):
  - Скасовано використання зворотної косої для метасимволів `{}` і `()`
  - Зворотна коса риска перед метасимволом скасовує його спеціальне значення
  - Додані метасимволи `+` `?` `|`

## Команда fgrep

- fgrep є більш швидким варіантом grep, оскільки НЕ розпізнає метасимволи регулярного виразу як спеціальні.
- Пошук буде завершено швидше, оскільки він обробляє тільки простий рядок, а не складний шаблон.

# Редагування файлів



## Командний режим

### Навігація

Спочатку натисніть  
ESC

**h** ← вліво  
**j** ↓ вниз  
**k** ↑ вгору  
**l** → вправо  
**G** на початок рядку  
**gg** на початок файлу

### пошук

/ пошук вперед  
? пошук назад

текстовий  
редактор  
**vi**

## Режим вставки

### Режим вставки

**a** почати введення з наступного символу  
**A** почати введення з кінця  
**i** почати введення з поточного символу  
**I** почати введення з початку рядка  
**o** створити рядок після поточної і почати введення  
**O** створити рядок перед поточною і почати введення

### Завершення роботи

Спочатку натисніть ESC

**ZZ** зберегти зміни і вийти  
**:w** зберегти зміни  
**:w!** примусово зберегти зміни  
**:q** закрити файл  
**:q!** закрити файл примусово, не зберігаючи зміни  
**:wq** зберегти зміни та вийти  
**:e** відкрити файл замість поточного  
**:e!** відкрити файл (зміни в поточному файлі не зберуться)  
**!:** запустити команду (наприклад ls, cp)  
**:x** зберегти зміни та вийти

### Вирізання, копіювання і вставка

Спочатку натисніть ESC

**cw** видалити поточне слово і почати введення  
**cc** видалити поточний рядок і почати введення  
**dd** вирізати поточний рядок  
**p** вставити після поточного символу  
**yy** скопіювати рядок  
**3yy** скопіювати наступні 3 рядки  
**x** видалити символ

## Редагування файлів

- Редактор ві входить в стандартну комплектацію майже кожної Linux системи.
- ві використовує два різних режими: командний режим і режим вставки.
- Для входу в режим вставки потрібно натиснути клавіші і, о або а.
- Для виходу з режиму вставки натисніть клавішу Esc.
- Для збереження файлу введіть :х або :qw, а для скасування змін і виходу :q!
- Для копіювання рядків або слів використовуйте у або уу, для вставки р.