УДК 004.45:004.65

Сокорчук І.П., ст. викладач
Харківський національний університет радіоелектроніки

ЕФЕКТИВНЕ ВИКОРИСТАННЯ СТАНДАРТНИХ УТИЛІТ LINUX ДЛЯ ПОШУКУ ДУБЛІКАТІВ ФАЙЛІВ

Пошук дублікатів файлів є актуальним завданням адміністрування комп'ютерних систем, оптимізації використання дискового простору та підвищення ефективності зберігання даних. Наявність дубльованих файлів призводить до нераціонального використання ресурсів, ускладнює резервне копіювання та знижує продуктивність обробки інформації. У зв'язку з цим важливим є застосування ефективних інструментів і методів для виявлення та аналізу однакових файлів у файлових системах.

Одним із найбільш гнучких підходів є використання стандартних засобів операційної системи Linux у поєднанні з командним інтерпретатором Bash. Зокрема, ефективним є комбінування утиліт find, xargs, md5sum, sha256sum, sort та uniq. Такий підхід дозволяє формувати масштабовані рішення без необхідності встановлення додаткового програмного забезпечення. Основна ідея полягає у генерації контрольних сум файлів і подальшому групуванні результатів для виявлення збігів.

Для автоматизації обробки великої кількості файлів доцільно застосовувати пакетну передачу аргументів через xargs. Ця утиліта забезпечує ефективне використання системних ресурсів за рахунок обробки даних блоками, а також підтримує обмежене розпаралелювання через параметр -P. Використання значення $(nproc) дозволяє задіяти всі доступні обчислювальні ядра, що суттєво підвищує продуктивність у багатоядерних системах. Разом із тим, xargs має обмежені можливості балансування навантаження і не забезпечує гнучкого керування процесами.

Альтернативним інструментом є утиліта GNU parallel, яка реалізує більш досконалі механізми паралельного виконання. Вона дозволяє автоматично розподіляти завдання між ядрами процесора, задавати кількість паралельних процесів і забезпечує впорядковане виведення результатів. Крім того, підтримується виконання завдань на віддалених вузлах, що розширює можливості застосування у розподілених системах. Недоліками є підвищене споживання ресурсів та необхідність попереднього встановлення.

Для формування контрольних сум можуть використовуватися алгоритми MD5, SHA-256 та SHA-512. Алгоритм MD5

характеризується високою швидкодією, проте має нижчу криптографічну стійкість. Алгоритми SHA-256 та SHA-512 забезпечують більшу надійність і точність ідентифікації файлів, хоча потребують більше обчислювальних ресурсів. Вибір конкретного алгоритму визначається співвідношенням вимог до швидкодії та достовірності результатів.

Спеціалізовані утиліти, такі як fdupes та rdfind, надають готові рішення для пошуку дублікатів. Вони поєднують використання контрольних сум із побайтовим порівнянням, що дозволяє зменшити кількість хибних збігів. Перевагою таких утиліт є простота використання, проте їх функціональність є менш гнучкою порівняно з комбінованими рішеннями на основі стандартних інструментів.

Додатковим підходом є застосування скриптів мовою Python із використанням бібліотеки hashlib. Це дозволяє реалізувати складні алгоритми фільтрації, оптимізації та обробки результатів. Такий підхід є доцільним у випадках, коли стандартні утиліти не забезпечують необхідної функціональності.

На рівні файлових систем також можливе використання вбудованих механізмів дедуплікації, зокрема у файловій системі btrfs. У цьому випадку усунення дублювання відбувається автоматично без участі користувача, що є ефективним для великих сховищ даних.

Таким чином, вибір методу пошуку дублікатів файлів залежить від обсягу даних, апаратних ресурсів та вимог до продуктивності. Для простих задач доцільно використовувати xargs у поєднанні зі стандартними утилітами. Для високонавантажених систем більш ефективним є застосування GNU parallel. У випадках, що потребують максимальної гнучкості, доцільно використовувати програмні рішення на Python або вбудовані можливості файлових систем.

Список використаних джерел:

1. GNU Findutils Manual [Електронний ресурс]. – Режим доступу: https://www.gnu.org/software/findutils/manual/ (дата звернення: 03.03.2025).

2. Kerrisk M. xargs(1) – Linux man-pages project [Електронний ресурс] / M. Kerrisk. – Режим доступу: https://man7.org/linux/man-pages/man1/xargs.1.html (дата звернення: 03.03.2025).

3. GNU Coreutils Manual [Електронний ресурс]. – Режим доступу: https://www.gnu.org/software/coreutils/manual/ (дата звернення: 03.03.2025).

4. GNU Parallel Manual [Електронний ресурс]. – Режим доступу: https://www.gnu.org/software/parallel/ (дата звернення: 03.03.2025).