

# Знайомство з інтерпретаторами команд POSIX. Командні файли sh і csh.

## Мета роботи

Дослідження можливостей командної оболонки в POSIX. Створення файлів налаштувань і засвоєння програмування в середовищі sh або csh.

## Основні положення

На поточний момент існує декілька варіантів командного інтерпретатора для ОС UNIX. Класичними (та стандартними) вважаються shell (автор - S. R. Bourne) і C-shell (автор - Bill Joy, Калифорнійський університет).

В таблиці нижче зібрані особливості найбільш популярних командних інтерпретаторів.

Функції, які підтримують різні командні оболонки

Назва	sh	cs	ks	ba	tc	zs	rc	es
Керування завданнями	N	Y	Y	Y	Y	Y	N	N

Розглянемо командний інтерпретатор на прикладі C-shell. Оболонка C-shell виконує дві основні функції: підтримка інтерактивного режиму роботи з ОС і програмування пакетного режиму виконання команд.

Як інтерпретатор, C-shell отримує команди користувача з терміналу або з простого командного файлу та породжує процеси, які відповідають командам. Як мова програмування C-shell містить змінні, керуючі оператори й оператори-модифікатори. Змінні в якості значень отримують слова та масиви слів.

Командний рядок складається зі списку слів і їх розділювачів.

Інтерпретатор також розрізняє метасимволи.

Розділювачі: ; ( ) < > & | <таб> <пробіл>

Метасимволи: ! \$ : \* , ? [ ] { } @ ~ . ^

Існує три різновиди лапок для використання в командах: подвійні, одинарні праві та одинарні ліві (прямі лапки ("), прямий апостроф (') і символ гравісу (')).

"рядок в подвійних лапках" інтерпретується завжди - в ньому можуть використовуватися метасимволи та виконуються підстановки значень змінних.

'рядок в одинарних правих лапках' - не інтерпретується та всі метасимволи не замінюються значеннями (виключення – метасимвол ! - він екранується зворотнім слешем \).

`рядок в одинарних лівих лапках` - інтерпретується як командний, виконується та замінюється результатом виконання.

Можна виконати будь яку послідовність команд, розділюючи їх символом ';' в одному командному рядку:

```
cd /mnt/stud; ls -l
```

Для керування виконанням послідовності команд використовуються логічні зв'язки '||', '&&' і круглі дужки (). Нехай A і B - команди ОС, тоді: A && B - команда B виконається, якщо A завершується успішно; A || B - команда B виконається, якщо A закінчиться з помилкою.

Для організації послідовної обробки потоку даних використовується можливість з'єднання стандартного виводу однієї команди зі стандартним вводом наступної - так звана конвеєризація. Для об'єднання команд в конвеєр вони розділюються символом '|':

```
ls -la | sort | more
```

В C-shell передбачена можливість виконання команд в асинхронному (фоновому) режимі, для цього в кінці команди ставлять символ '&':

```
ls -l > /dev/lp&
```

Для перенаправлення потоків вводу/виводу використовують такі розділювачі:

```
> , < , >> , << , >& , >>&
```

Команда ОС 'tee' дублює стандартний вивід.

В мові C-shell визначені такі типи змінних: - слово - рядок - масив слів - позиційна змінна.

## Порядок виконання роботи

**Створити або модифікувати файли .cshrc, .login и .logout (або відповідні для bash) в своєму реєстраційному каталозі.**

**Розробити та зневадити командний файл відповідно до заданого варіанту. Програми мають містити діагностику помилок запуску та виконання.**

## Варіанти завдань

№1. Написати командний файл, який з заданого в командному рядку файлу обирає слова, які починаються з великої букви та записує їх в файл "Upper" в поточному каталозі, а всі слова, які починаються з маленьких букв - в файл "Lower", інші слова записуються в файл "Misc". Потім всі файли сортуються та збираються в один - кожен джерельний файл в окремій колонці. (Можна використовувати команди grep, fgrep, sort и pr).

№2. Командний файл, який перемещує з поточного каталогу всі файли з заданими в командному рядку розширеннями в підкаталоги поточного каталогу з іменем, яке співпадає з розширенням. Якщо такий підкаталог відсутній, то його потрібно створити. Якщо нема файлів, які відповідають якомусь розширенню - видати повідомлення.

№3. Командний файл, який з заданих в командному рядку каталогів рекурсивно знищує всі файли з розширенням ".o" та файли "a.out", запитуючи підтвердження, а файли з розширенням .c переносить в підкаталог заданого каталогу з іменем C\_src. При цьому необхідно видавати повідомлення про переміщення по дереву каталогів і про переміщення файлів.

№4. Командний файл, в якому переглядаються всі підкаталоги поточного каталогу та створюються два списки каталогів. Якщо в підкаталозі, що переглядається, кількість

файлів менше 10, то список цих файлів з рядком-заголовком - повним іменем підкаталогу, дописують в файл "Littlelist", інакше - в "Biglist".

№5. Командний файл, який імена всіх файлів символічних пристроїв з вказаного в командному рядку каталогу, імена яких починаються з символу "r", записує в файл "chardevice", а тих, що починаються з "t" - в файл "ttydevice". Кожен файл передуюється заголовком і вони виводяться на екран одночасно, кожен в своїй колонці (див. команду "pr").

№6. Командний файл, який в каталозі, ім'я якого вказано першим параметром командного рядку, вибирає файли, власник яких вказаний другим параметром команди, і сортує їх по ключу, вказаному третім параметром командного рядку. Потрібно видавати діагностичні повідомлення у випадку відсутності каталогу, файлів в каталозі або невірному використанні ключу. (Пропонується використовувати команди sort, ls -l та змінні типу масив слів).

№7. Командний файл, який імена всіх файлів поточного каталогу збирає в два файли. В перший файл з іменем "myfile" всі файли, власником яких є поточний користувач (тобто ім'я, з яким увійшли в систему), а в інший файл (з іменем "other") - імена решти файлів. Кожен файл передувати заголовком і вивести одночасно в дві колонки (по одному файлу в колонці), відсортувавши по іменах.

№8. Командний файл, який сортує всі файли поточного каталогу за кількістю рядків і записує їх імена в файл "sort\_line", за кількістю слів - в файл "sort\_word" і за кількістю символів - в файл "sort\_char". (Пропонується використовувати команду wc і змінні типу масив слів).

№9. Командний файл, який з каталогу, ім'я якого вказано в командному рядку першим параметром (якщо такого каталогу нема, то видати діагностичне повідомлення), вибирає імена файлів, які починаються або з латинської літери, або з іншого символу в залежності від ключа, який заданий другим параметром командного рядка (якщо другого параметру нема, то видати діагностичне повідомлення), сортує їх за часом створення та видає їх пронумерований список в файл "my\_sort".

№10. Командний файл, який підраховує в каталогах, які задані аргументом командного рядку, загальну кількість файлів, загальну кількість рядків, слів і символів у всіх файлах. Якщо вказаний каталог не існує, то видати діагностичне повідомлення.

№11. Командний файл, який в каталозі, заданому першим аргументом командного рядку, вибирає з усіх файлів рядки, в яких є слова, що задані в якості шаблону другим аргументом командного рядку. Видати діагностичне повідомлення, якщо нема заданого шаблону або нема каталогу; виводити імена файлів, в яких є слова, що шукаються.

№12. Командний файл, який в поточному каталозі залишає тільки файли, що були створені користувачем, який працює зараз в системі (\$user), решту розміщує в підкаталог поточного з іменем "other" (якщо файлів нема, то видати діагностичне повідомлення, якщо підкаталогу не існувало, то створити та вивести діагностику). Потім вивести повний список поточного каталогу та підкаталогу "other".

## Вміст звіту

Вміст файлів .login, .cshrc и .logout (або відповідних для командної оболонки, що використовується) і текст командного файлу.

У звіті мають міститися відповіді на питання пункту "Порядок виконання роботи": Короткий опис кожної операції, що досліджується, та команди інтерпретаторів, що

використовуються, результати дій зі змінними.

## Контрольні питання

1. Призначення внутрішніх змінних і способи зміни їх значень.
2. Що таке стандартний ввід і вивід, способи їх перенаправлення.
3. Операції в мові C-shell.
4. Керування послідовністю виконання команд в мові C-shell.
5. Передача параметрів в командні файли та процедури.
6. Межі дії (область видимості) змінних.
7. Оператори циклу, оператори умови та вибору.
8. Зовнішні змінні C-shell.
9. Змінні оточення: створення, час життя та межі дії.
10. Керування процесами в C-shell.
11. Перенаправлення вводу, виводу та виводу помилок в bourne shell.
12. Передача на термінал керуючих символів в shell і C-shell.
13. Ввід з терміналу в командному файлі в shell і C-shell.