

ПРАКТИЧНА РОБОТА 02
Сценарії командного рядка bash
План

1. Відсутні бібліотеки.
2. Удосконалення користувацьких команд.
3. Створення утиліт.

1. Відсутні бібліотеки

Розробіть відповідно до представлених задач сценарії командного рядка.
Доповніть кожний сценарій виводом автора скрипту:

- 1) ***Пошук програм у PATH.*** Перевірити доступність вказаної програми у списку шляхів PATH – допустимість шляхів до файлів.
- 2) ***Перевірка вводу щодо обмеження на алфавітно-цифрові символи.*** Чи введені символи відносяться тільки до цифр і букв верхнього/нижнього регістрів?
- 3) ***Нормалізація форматів дат.*** Сценарій повинен нормалізувати рядки з датами відносно простого набору критеріїв: місяць повинен задаватись назвою або числом в діапазоні від 1 до 12, а рік – чотиризначним числом.
- 4) ***Зручне для читання представлення великих чисел.*** Сценарій повинен відображати передане число в форматі представлення з комами. Передбачається наявність змінних-роздільників для дробової частини та груп розрядів.
- 5) ***Перевірка вводу на цілі числа.*** Процедура повинна оцінювати правильність форматування від'ємних чисел та входження значень у деякий встановлений користувачем діапазон.
- 6) ***Перевірка вводу на дійсні числа.*** Сценарій повинен перевіряти дробове число шляхом його розбиття на 2 частини: цілу та дробову. Перша частина перевіряється як звичайне ціле число, а друга – як додатне ціле число. Сценарій не розпізнає наукову форму запису числа.
- 7) ***Перевірка форматів дат.*** Необхідно перевіряти допустимість введених дат разом з урахуванням високосності року.
- 8) ***Покращення неякісних реалізацій echo.*** Сценарій передбачає визначення того, яка версія echo використовується в системі (Unix чи GNU/Linux). Залежно від підтримки прапорця -n слід придушувати спеціальний символ '\c', який додається автоматично в кінець введеного рядка.
- 9) ***Обчислення довільної точності з дійсними числами.***

- 10) **Блокування файлів.** Файли-блокування виступають в ролі семафорів для сценаріїв взаємодії з файлами зі спільним доступом. Сценарій запитує наявність файлу-блокування, а після видалення такого файлу спільний файл можна використовувати іншим процесам.
- 11) **ANSI-послідовності управління кольором.** Пропонується використовувати спеціальні змінні для форматування тексту. Наприклад, назви змінних, які закінчуються на 'f' (foreground), відповідають кольорам шрифту, а суфікс 'b' (background) вказує на колір фону.
- 12) **Створення бібліотечних сценаріїв.** Для того, щоб перетворити раніше розроблені функції в бібліотеку для використання в інших сценаріях, слід витягти всі функції та необхідні глобальні змінні або масиви та помістити їх в один великий файл.
- 13) **Налагодження сценаріїв.** Для трасування змінних можна використовувати команду echo, а також запускати сценарії командою bash -x. Розглянемо налагодження на прикладі простої гри «вгадай число»

```
biggest=100
guess=0
guesses=0
number=$(( $$ % $biggest )      # Випадкове число
                                   # від 1 до $biggest
echo "Guess a number between 1 and $biggest"
while [ "$guess" -ne $number ] ; do
    /bin/echo -n "Guess? " ; read answer
    if [ "$guess" -lt $number ] ; then
        echo "... bigger!"
    elif [ "$guess" -gt $number ] ; then
        echo "... smaller!"
    fi
    guesses=$(( $guesses + 1 ))
done

echo "Right!! Guessed $number in $guesses guesses."
exit 0
```

2. Удосконалення користувацьких команд

- 14) **Форматування довгих рядків.** У деяких системах відсутня команда `fmt` для форматування текстів великої довжини (перенесення рядків).
- 15) **Резервне копіювання файлів при видаленні.** Сценарій має дозволяти просту можливість відновлення, створюючи та використовуючи новий каталог у домашньому каталозі користувача. Якщо користувач додаватиме прапорець `-f`, файли видаляються БЕЗ архівування.
- 16) **Робота з архівом видалених файлів.** Сценарій має шукати в архіві видалених файлів потрібний файл або каталог. Якщо знайдено більше одного співпадіння, слід вивести перелік результатів пошуку, упорядкований за часом, а також запропонувати користувачеві потрібний з них для відновлення.
- 17) **Журналювання операцій видалення файлів.** Потрібно журналювати виклики команди `rm` в окремому файлі, не сповіщаючи про це користувача.
- 18) **Вивід вмісту каталогів.** Сценарій повинен генерувати багатостовпцевий список файлів та каталогів, показуючи розміри для файлів та кількість записів для каталогів.
- 19) **Пошук файлів за іменами.** Сценарій має створювати «базу даних» за допомогою команди `find`. Для запуску потрібно володіти root-правами.
- 20) **Імітація інших середовищ: MS-DOS.** Використовуючи аліаси, зімітуйте `DIR`-команду DOS у командному рядку `bash`.
- 21) **Вивід часу в різних часових поясах.** Сценарій повинен виводити поточний час у вказаному часовому поясі або географічній області. При виклику без аргументів має виводитись час UTC/GMT. Використовуйте ключове слово `list` для виводу списку всіх відомих географічних областей.

3. Створення утиліт

- 22) **Утиліта для нагадування.** Сценарії `remember` та `remindme` повинні дозволяти зберігати короткі замітки та виводити нагадування на екран відповідно.
- 23) **Інтерактивний калькулятор.** Напишіть сценарій-обгортку для задачі 9, який додасть інтерактивності калькулятору командного рядка. Не забудьте помістити сценарій задачі 9 в один з каталогів зі списку `PATH`.

- 24) ***Перетворення температур.*** Сценарій має дозволяти вводити температуру в градусах Фаренгейта, Цельсія або Кельвіна та отримувати еквівалентну температуру в інших двох шкалах.
- 25) ***Обчислення платежів за кредитом.***
- 26) ***Стеження за подіями.*** Реалізувати просту програму-календар з 2 сценаріїв. Перший з них дозволить визначити подію, яка буде одноразовою або повторюватиметься. Усі дати перевірятимуться та зберігатимуться разом з однорядковим описом події в спеціальному файлі. Другий сценарій переглядатиме усі збережені події та шукатиме заплановані на поточну дату.