Завдання

Домашне завдання No2

«Взаємодія із середовищем виконання»
Написати програму environ, що приймає короткі та довгі опції і виконує наступні завдяня:

- якщо програма запущена без опцій, то в стандартний потік виводу выводиться інформація о поточном оточенні;
- якщо вказана коротка опція -h (або довга --help), то в стандартний потік выводу буде выведено інформацию по роботі із програмою;
- якщо вказана коротка опція -i <змінна> (або довга --info <змінна>), то в стандартний потік выводу буде выведено значення вказаної змінної або повідомлення про те, що вказаної змінної в оточенні немає;
- якщо вказана коротка опція -s <змінна=значення> (або довга --set <змінна=значення>), то вказана змінна оточення отримає нове значення та в стандартний потік виводу буде виведено встановлене значення вказаної змінної;
- якщо вказані короткі опції -a <змінна> та -v <значення> (або довгі --assign <змінна> та --value <значення>), то вказана змінна оточення отримає нове
- значення та в стандартний потік виводу буде виведено встановлене значення вказаної змінної;
- якщо значення не вказано, то буде присвоєно пустий рядок; якщо же не вказано змінну, то присвоєння не відбудається, а в стандартний потік помилок виводиться відповідне повідомлення;
- якщо вказана коротка опція -d <змінна> (або довга --del <змінна>), то вказана змінна видаляєтья з оточення;
- якщо вказана коротка опція -c (або довга --clear), то програма повністю очищує оточення.

Кожна дія має бути оформлена у вигляді функції; функції, що виконують відповідні дії, мають бути зібрані в окремому вихідному файлі. Підготувати Makefile для збирання програми.

Розв’язок

Запустимо нашу програму environ, вона виведе всі поточні змінні оточення.

mint@asus-mint:~/git/op-course/lab2/bin$ ./environ

SHELL=/bin/bash

SESSION_MANAGER=local/asus-mint:@/tmp/.ICE-unix/5663,unix/asus-mint:/tmp/.ICE-unix/5663

QT_ACCESSIBILITY=1

COLORTERM=truecolor

XDG_CONFIG_DIRS=/etc/xdg/xdg-cinnamon:/etc/xdg

XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0

GNOME_DESKTOP_SESSION_ID=this-is-deprecated

LANGUAGE=en_US

LC_ADDRESS=uk_UA.UTF-8

LC_NAME=uk_UA.UTF-8

SSH_AUTH_SOCK=/run/user/1000/keyring/ssh

DESKTOP_SESSION=cinnamon

LC_MONETARY=uk_UA.UTF-8

GTK_MODULES=gail:atk-bridge

XDG_SEAT=seat0

PWD=/home/mint/git/op-course/lab2/bin

LOGNAME=mint

XDG_SESSION_DESKTOP=cinnamon

QT_QPA_PLATFORMTHEME=qt5ct

XDG_SESSION_TYPE=x11

GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1

XAUTHORITY=/home/mint/.Xauthority

XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/mint

GDM_LANG=en_US

INSIDE_NEMO_PYTHON=

…

Запустимо ./bin/environ --help. Виведе невеличку інформацію про команди

mint@asus-mint:~/git/op-course/lab2$ ./bin/environ --help

Usage: environ [OPTIONS]

Options:

-h, --help Show this help message

-i, --info <var> Show the value of the specified environment variable

-s, --set <var=val> Set the value of the specified environment variable

-a, --assign <var> Assign a value to the specified environment variable

-v, --value <val> Value to assign when using --assign

-d, --del <var> Delete the specified environment variable

-c, --clear Clear all environment variables

Отримаємо значення змінної оточення. Для виведення значення змінної PATH використовуйте опцію -i або --info.

mint@asus-mint:~/git/op-course/lab2$ ./bin/environ --info PATH

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin

Якщо змінної не існує:
mint@asus-mint:~/git/op-course/lab2$ ./bin/environ --info NON_EXISTENT_VAR
Variable NON_EXISTENT_VAR not found in environment.

Щоб встановити нове значення змінної, можна використовувати опцію -s або --set.
mint@asus-mint:~/git/op-course/lab2$ ./bin/environ --set MY_VAR=hello
Set MY_VAR=hello

Зараз виведемо її:
mint@asus-mint:~/git/op-course/lab2$ ./bin/environ --info MY_VAR
Variable MY_VAR not found in environment.

Чому же так?

Зазвичай змінна середовища зберігається тільки під час виконання програми. Ця змінна буде існувати тільки в контексті цієї програми або під час її роботи. Як тільки програма завершує виконання, змінна оточення зникає. Змінна MY_VAR не буде знайдена, тому що змінні середовища встановлені у попередній сесії, більше не існують.

Що робити, щоб зберегти змінну середовища?

Щоб змінна була доступною в наступних командах, ми повинні встановити її у тому ж середовищі, яке виконує ці команди. Наприклад, можна встановлювати змінні ось так:

export MY_VAR=hello

Якщо ми хочете зберегти змінні для подальшого використання під час виконання програми, необхідно використовувати змінні середовища у межах цієї програми або зберігати їх в системних файлах конфігурацій, наприклад, .bashrc або .profile.

Давайте встановимо нову змінну.

mint@asus-mint:~/git/op-course/lab2$ export MY_VAR=hello

./bin/environ --info MY_VAR

І вже бачимо:

mint@asus-mint:~/git/op-course/lab2$ ./bin/environ --info MY_VAR

MY_VAR=hello

Все вийшло!

Давайте видалимо нашу створену змінну

mint@asus-mint:~/git/op-course/lab2$ ./bin/environ --del MY_VAR

Deleted variable: MY_VAR

mint@asus-mint:~/git/op-course/lab2$ ./bin/environ --info MY_VAR

MY_VAR=hello

І знову ми бачимо проблему! Основне питання тут у тому, що змінні середовища в Linux системах є специфічними для кожного процесу і не можуть впливати на оболонку. Переглянемо наші дії:

1. Виконуємо export MY_VAR=hello, і ця змінна існує в середовищі вашої оболонки.
2. Далі запускаємо ./bin/environ --del MY_VAR, що видаляє змінну лише в контексті програми, але не в середовищі оболонки.
3. Після завершення програми ми знову запитуємо./bin/environ --info MY_VAR, і оболонка все ще бачить змінну, тому що її середовище не змінилося.

На жаль, програмно змінювати змінні середовища у оболонці Bash безпосередньо не можна. Це обмеження всіх процесів у Unix-подібних системах. Однак можна встановлювати і видаляти змінні в оболонці за допомогою спеціальних команд в самому Bash:

unset MY_VAR

Якщо ж нам потрібно використовувати програму для встановлення змінних середовища в оболонці, можна запускати її з попередньо визначеними командами в тому ж середовищі за допомогою наступного синтаксису:

export $(./bin/environ --set MY_VAR=hello)

unset $(./bin/environ --del MY_VAR)

І тоді вже матимемо наступно ситуацію:

mint@asus-mint:~/git/op-course/lab2$ unset MY_VAR

mint@asus-mint:~/git/op-course/lab2$ ./bin/environ --info MY_VAR

Variable MY_VAR not found in environment.

mint@asus-mint:~/git/op-course/lab2$ export $(./bin/environ --set MY_VAR=hello)

mint@asus-mint:~/git/op-course/lab2$ ./bin/environ --info MY_VAR

MY_VAR=hello

mint@asus-mint:~/git/op-course/lab2$ unset $(./bin/environ --del MY_VAR)

mint@asus-mint:~/git/op-course/lab2$ ./bin/environ --info MY_VAR

Variable MY_VAR not found in environment.

Наша проблема вирішена.

Давайте знову оглянемо нашу проблему. Ось ключові моменти, які допоможуть зрозуміти проблему:

1. **Змінні середовища не можуть змінюватися у батьківському процесі**: Коли ми запускаємо програму, вона виконується як окремий процес. Змінні середовища, встановлені або видалені всередині цього процесу, діють тільки в межах самого процесу програми. Як тільки програма завершується, всі зміни (включаючи видалення змінних середовища) зникають разом з процесом програми.
   Тобто, коли ми робимо export MY_VAR=hello у Bash, змінна встановлюється у середовищі оболонки (Bash). Але коли запускаємо програму ./bin/environ --del MY_VAR, ця команда створює новий процес, і видалення змінної виконується тільки в цьому процесі. Після завершення програми повертаємося до початкового середовища оболонки, де змінна все ще існує.

2. **Зміни середовища обмежені процесом програми**: Операція видалення змінної --del або unsetenv() працює тільки в рамках самої програми і не впливає на зовнішні процеси, включаючи оболонку (Bash). Тому, коли ми запускаємо програму з опцією --del, вона дійсно видаляє змінну в її локальному середовищі, але це середовище зникає разом із завершенням процесу.

Висновок

Під час виконання цього завдання була розроблена програма environ, яка може працювати з різними змінними середовища за допомогою коротких та довгих опцій командного рядка. Програма виконує наступні дії:

1. Виводить усі змінні середовища.
2. Надає інформацію про вказану змінну середовища.
3. Встановлює або змінює значення змінної.
4. Видаляє змінні з середовища.
5. Очищує все середовище.

Під час тестування функціональності було виявлено важливе обмеження: зміни змінних середовища (встановлення, видалення) в Unix-подібних системах можуть відбуватися лише в межах самого процесу програми. Іншими словами, змінні середовища, які змінюються в процесі виконання програми, не впливають на батьківський процес (наприклад, оболонку Bash), з якого програма була запущена. Після завершення програми всі зміни в змінних середовища зникають.

Основні труднощі, з якими ми зіткнулися:

- Неможливість збереження змінних середовища після завершення програми.
- Неможливість видалення змінних середовища у батьківському процесі.

Було зроблено висновок, що для встановлення або видалення змінних в оболонці необхідно використовувати системні команди оболонки, такі як export та unset. Програма може використовуватися для

роботи зі змінними середовища всередині одного процесу або в рамках сценаріїв, де важлива робота з оточенням у межах самої програми.

Таким чином, програма environ є ефективним інструментом для маніпулювання змінними середовища в межах одного виконання, проте для тривалих змін у зовнішньому середовищі слід використовувати вбудовані механізми оболонки.