

Харківський національний університет імені В.Н. Каразіна
Факультет комп'ютерних наук

Звіт
з лабораторної роботи №2
дисципліна: «Операційні системи»

Виконала: студентка групи КС34
Євстігнєєва Марія Олексіївна
Перевірів: Паршинцев Богдан Володимирович

Харків 2022

«Взаимодействие со средой исполнения»

Написать программу `environ`, которая принимает короткие и длинные опции и выполняет следующие задачи:

- ♦ если программа вызвана без опций, то в стандартный поток вывода выводится информация о текущем окружении;
- ♦ если указана короткая опция `-h` (или длинная `--help`), то в стандартный поток вывода будет выведена информация по работе с программой;
- ♦ если указана короткая опция `-i <переменная>` (или длинная `--info <переменная>`), то в стандартный поток вывода будет выведено значение указанной переменной или сообщение о том, что данной переменной в окружении нет;
- ♦ если указана короткая опция `-s <переменная=значение>` (или длинная `--set <переменная=значение>`), то указанная переменная окружения получит новое значение и в стандартный поток вывода будет выведено установленное значение указанной переменной;
- ♦ если указана короткая опция `-a <переменная>` и `-v <значение>` (или длинная `--assign <переменная>` и `--value <значение>`), то указанная переменная окружения получит новое значение и в стандартный поток вывода будет выведено установленное значение указанной переменной; если значение не указано, то присваивается пустая строка; если переменная не указана, то присваивание не производится и в стандартный поток ошибок выводится соответствующее сообщение;
- ♦ если указана короткая опция `-d <переменная>` (или длинная `--del <переменная>`), то указанная переменная удаляется из окружения;
- ♦ если указана короткая опция `-c` (или длинная `--clear`), то программа полностью очищает окружение.

Каждое действие должно быть оформлено в виде функции; функции, выполняющие действия, должны быть собраны в отдельном исходном файле. Подготовить `Makefile` для сборки программы.

Результати:

```
Environment: 83
72
69
76
76
61
47
98
105
110
47
98
97
115
104
```

```

1) якщо програма запущена без опцій, то в стандартний потік виводу виводиться інформація о поточн
ому оточенні
2) якщо вказана коротка опція - h або довга -- help, то в стандартний потік виводу буде виведено
інформацію по роботі із програмою
3) якщо вказана коротка опція - i <змінна> або довга -- info <змінна>, то в стандартний потік вив
оду буде виведено значення вказаної змінної або повідомлення про те, що вказаної змінної в оточен
ні немає
4) якщо вказана коротка опція - s <змінна=значення> або довга -- set <змінна=значення>, то вказан
а змінна оточення отримає нове значення та в стандартний потік виводу буде виведено встановлене з
начення вказаної змінної
5) якщо вказані короткі опції - a <змінна> та - v <значення> або довгі -- assign <змінна> та -- v
alue <значення>, то вказана змінна оточення отримає нове значення та в стандартний потік виводу б
уде виведено встановлене значення вказаної змінної; якщо значення не вказано, то буде присвоєно п
устий рядок; якщо же не вказано змінну, то присвоєння не відбувається, а в стандартний потік поми
лок виводиться відповідне повідомлення
6) якщо вказана коротка опція - d <змінна> або довга -- del <змінна>, то вказана змінна видаляєть
ся з оточення
7) якщо вказана коротка опція - c або довга -- clear, то програма повністю очищує оточення

```

```
The value of variable: 12
```

```
The value of variable: 12
```

```
Now the value of variable: 10
```

```
Now the value of variable: 5
```

```
Now the value of variable: 0
```

```
The variable with value 12 was deleted
```

```

The variable with value 2 was deleted
The variable with value 99 was deleted
The variable with value 12 was deleted
The variable with value 83 was deleted
The variable with value 72 was deleted
The variable with value 69 was deleted
The variable with value 76 was deleted
The variable with value 76 was deleted
The variable with value 61 was deleted
The variable with value 47 was deleted
The variable with value 98 was deleted
The variable with value 105 was deleted
The variable with value 110 was deleted
The variable with value 47 was deleted
The variable with value 98 was deleted
The variable with value 97 was deleted
The variable with value 115 was deleted
The variable with value 104 was deleted

```

Код:

```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <getopt.h>

```

```

#include <string.h>
#include "func.h"

struct map {
    char name[20];
    int* adress;
}map[20];

char assignV[20];
int valueV;

int actualSize = 0; //actual size of the map

int main(int argc, char const *argv[])
{
    extern char **environ;
    strcpy(map[0].name, "argc"); //adding new element to the map
    map[0].adress = &argc;
    actualSize++;

    if(argc == 1) {
        printf("\nEnvironment: ");
        while(**environ != NULL) {
            printf("%d", **environ);
            printf("\n");
            *environ += 1;
        }
        return 0;
    }
}

```

```

const struct option long_opt[] = {
    {"help", 0, 0, 'h'},
    {"info", 1, 0, 'i'},
    {"set", 1, 0, 's'},
    {"assign", 1, 0, 'a'},
    {"value", 1, 0, 'v'},
    {"del", 1, 0, 'd'},
    {"clear", 1, 0, 'c'},
    {NULL, 0, NULL, 0}
}; //array of options

```

```

int opt;
strcpy(map[1].name, "opt");
map[1].adress = &opt;
actualSize++;

```

```

int variable = 12;
strcpy(map[2].name, "variable"); //adding new element to the map
map[2].adress = &variable;
actualSize++;

```

```

int flA = 0, flV = 0;
while((opt = getopt_long(argc, argv, "hi:s:a:v:d:c", long_opt, NULL)) != -1) {
    switch(opt) {
        case 'h':
            help();
            break;
        case 'i':

```

```

info();
break;
case 's':
    printf("\n");
    char nameVar[20]; //name of variable
    char val[10]; //value of variable
    char nameWritten[30]; //what user entered
    strcpy(nameWritten, optarg);
    for(int i = 0; i < strlen(nameWritten); i++) {
        if(nameWritten[i] == '=') break;
        nameVar[i] = nameWritten[i];
    }
    for(int i = 0; i < strlen(nameWritten) - strlen(nameVar) - 1; i++) {
        val[i] = nameWritten[i + strlen(nameVar) + 1];
    }
    int fl = 0;
    for(int i = 0; i < actualSize; i++) {
        if(strcmp(map[i].name, nameVar) == 0) {
            *(map[i].adress) = atoi(val);
            printf("Now the value of variable: %d", *(map[i].adress));
            printf("\n");
            fl = 1;
        }
    }
    if(!fl) printf("There is no variable with such name!\n");
    break;
case 'a':
    flA = 1;
    strcpy(assignV, optarg);

```

```
break;
case 'v':
    flV = 1;
    valueV = atoi(optarg);
    break;
case 'd':
    del();
    break;
case 'c':
    clear();
    break;
}
}
if(flV && flA) AssignValue();
else if(flV && !flA) value();
else if(!flV && flA) assign();
return 0;
}
```

```
void help(){
    printf("\n");
    printf("1) якщо програма запущена без опцій, то в стандартний потік виводу
виводиться інформація о поточному оточенні\n");
    printf("2) якщо вказана коротка опція - h або довга -- help, то в стандартний
потік виводу буде виведено інформацію по роботі із програмою\n");
    printf("3) якщо вказана коротка опція - i <змінна> або довга -- info <змінна>,
то в стандартний потік виводу буде виведено значення вказаної змінної або
повідомлення про те, що вказаної змінної в оточенні немає\n");
```

printf("4) 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\n");

printf("5) 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 відбудеться, а в стандартний потік помилок виводиться відповідне повідомлення\n");

printf("6) якщо вказана коротка опція - d <змінна> або довга -- del <змінна>, то вказана змінна видаляється з оточення\n");

printf("7) якщо вказана коротка опція - c або довга -- clear, то програма повністю очищує оточення\n");

}

void info(){

printf("\n");

int fl = 0;

for(int i = 0; i < actualSize; i++) {

if(strcmp(map[i].name, optarg) == 0) {

printf("The value of variable: %d", *(map[i].adress));

printf("\n");

fl = 1;

}

}

if(!fl) printf("There is no variable with such name!\n");

}


```

void assign(){
    int fl = 0;
    for(int i = 0; i < actualSize; i++) {
        if(strcmp(map[i].name, assignV) == 0) {
            *(map[i].adress) = 0;
            printf("Now    the    value    of    variable:    %d",
*(map[i].adress));

            printf("\n");
            fl = 1;
        }
    }
    if(!fl) printf("There is no variable with such name!\n");
}

```

```

void value(){
    printf("\nYou didn't enter the variable!\n");
}

```

```

void AssignValue(){
    int fl = 0;
    for(int i = 0; i < actualSize; i++) {
        if(strcmp(map[i].name, assignV) == 0) {
            *(map[i].adress) = valueV;
            printf("Now    the    value    of    variable:    %d",
*(map[i].adress));

            printf("\n");
            fl = 1;
        }
    }
}

```

```

    }
    if(!fl) printf("There is no variable with such name!\n");
}

void del(){
    printf("\n");
    int fl = 0;
    for(int i = 0; i < actualSize; i++) {
        if(strcmp(map[i].name, optarg) == 0) {
            int var = *(map[i].adress);
            map[i].adress = NULL;
            strcpy(map[i].name, "\0");
            printf("The variable with value %d was deleted", var);
            printf("\n");
            fl = 1;
        }
    }
    if(!fl) printf("There is no variable with such name!\n");
}

```

```

void clear(){
    for(int i = 0; i < actualSize; i++) {
        int var = *(map[i].adress);
        map[i].adress = NULL;
        strcpy(map[i].name, "\0");
        printf("The variable with value %d was deleted", var);
        printf("\n");
    }
    extern char **environ;
}

```

```
while(**environ != NULL) {  
    printf("The variable with value %d was deleted", **environ);  
    printf("\n");  
    *environ += 1;  
}  
}
```

Хедер файл:

```
#define func.h
```

```
void help();  
void info();  
void assign();  
void value();  
void AssignValue();  
void del();  
void clear();
```

Makefile:

```
lab2: lab2.o
```

```
lab2.o: lab2.c func.h
```

```
gcc -c lab2.c
```