Operating Systems. Homework 2

Operating Systems

Александр Васюков | БПИ235

Задание

`Отыскать и привести корректно выполняемые примеры не менее пяти скриптов, использующих следующие инструкций языка bash: if; while; использование функций на bash, вызываемых из других функций. Каждый из файлов примеров снабдить своими комментариями, поясняющими, что данный скрипт делает.`

Решение

Более удобно смотреть скрипты и отчёт на гитхабе по ссылке:
https://github.com/vasyukov1/HSE-FCS-SE-2-year/tree/main/Operating%20Systems/Homeworks/02

1. find_file.sh

   Скрипт проверяет, существует файл или нет.

   $ find_file.sh

   ```
   #!/bin/bash
   # Check file existence.

   if [ -e "$1" ]; then
       echo "File '$1' exists."
   else
       echo "File '$1' doesn't exist."
   fi
   ```

   Результат:

   ```
   alexvasyukov@Alexanders-MacBook-Air 02 % chmod +x find_file.sh
   alexvasyukov@Alexanders-MacBook-Air 02 % ./find_file.sh find_file.sh
   File 'find_file.sh' exists.
   alexvasyukov@Alexanders-MacBook-Air 02 % ./find_file.sh phantom_file.txt
   File 'phantom_file.txt' doesn't exist.
   ```

2. square.sh

   Обратный отсчёт от числа, которое ввёл пользователь, а также вывод квадратов чисел.

```
$ square.sh
#!/bin/bash
# Outputs a countdown of squares of numbers, starting from the number entered by the user.

number=$1

if ! [[ $number =~ ^[0-9]+$ ]]; then
    echo "Please, send the positive number."
    exit 1
fi

while [ $number -ge 0 ]; do
    sq=$(echo $number^2 | bc)
    echo "$number^2 = $sq"
    number=$((number - 1))
    sleep 1
done

echo "Countdown is done!"
```

Проверяется, что введено положительное число.

В переменную `sq` записывается квадрат числа.

Переменная `number` уменьшается на 1.

`sleep 1` для вывода чисел через 1 секунду.

Результат:

```
alexvasyukov@Alexanders-MacBook-Air 02 % chmod +x square.sh
alexvasyukov@Alexanders-MacBook-Air 02 % ./square.sh 10
10^2 = 100
9^2 = 81
8^2 = 64
7^2 = 49
6^2 = 36
5^2 = 25
4^2 = 16
3^2 = 9
2^2 = 4
1^2 = 1
0^2 = 0
Countdown is done!
```

3. fibonacci.sh

   Считает число Фибоначчи, которое введёт пользователь.

```
$ fibonacci.sh
#!/bin/bash
# Count the Fibonacci number which enter the user.

if [ -z "$1" ]; then
    echo "You need to enter the number!"
    exit 1
fi

n=$1

fibonacci() {
    local num=$1
    if [ "$num" -eq 0 ]; then
        echo 0
    elif [ "$num" -eq 1 ]; then
        echo 1
    else
        local a=$(fibonacci $((num - 1)))
        local b=$(fibonacci $((num - 2)))
        echo $((a + b))
    fi
}

result=$(fibonacci $n)
echo "F($n) = $result"
```

В рекурсивной функции проверяются базовые условия, когда число равно `0` или `1`. Затем в локальные переменные записываются результаты вычисления предыдущих двух чисел.

Результат:

```
alexvasyukov@Alexanders-MacBook-Air 02 % chmod +x fibonacci.sh
alexvasyukov@Alexanders-MacBook-Air 02 % ./fibonacci.sh 7
F(7) = 13
alexvasyukov@Alexanders-MacBook-Air 02 % ./fibonacci.sh 10
F(10) = 55
alexvasyukov@Alexanders-MacBook-Air 02 %
```

4. menu.sh
   Предлагает пользователю выбрать блюда, а в конце считает итоговый счёт за все блюда.

$ menu.sh

```bash
#!/bin/bash
# Counter of the bill.

bill=0

print_menu() {
    echo "Choose the dish:"
    echo "1. Salad"
    echo "2. Borsch"
    echo "3. Pasta"
    echo "4. Dessert"
    echo "0. Check the bill."
}

add_salad() {
    salad=1.5
    bill=$(echo "scale=2; $bill + $salad" | bc)
    echo "Salad: $salad euro"
}

add_borsch() {
    borsch=0.5
    bill=$(echo "scale=2; $bill + $borsch" | bc)
    echo "Borsch: $borsch euro."
}

add_pasta() {
    pasta=2
    bill=$(echo "scale=2; $bill + $pasta" | bc)
    echo "Pasta: $pasta euro."
}
```

```
32
33  add_desert() {
34      dessert=1
35      bill=$(echo "scale=2; $bill + $dessert" | bc)
36      echo "Dessert: $dessert euro."
37  }
38
39  get_bill() {
40      echo "The bill is $bill euro."
41  }
42
43  while true; do
44      print_menu
45      read -p "Enter the number of option: " choice
46      if [ "$choice" -eq 1 ]; then
47          add_salad
48      elif [ "$choice" -eq 2 ]; then
49          add_borsch
50      elif [ "$choice" -eq 3 ]; then
51          add_pasta
52      elif [ "$choice" -eq 4 ]; then
53          add_desert
54      elif [ "$choice" -eq 0 ]; then
55          get_bill
56          break
57      else
58          echo "Use correct options. From 0 to 4."
59      fi
60      echo
61  done
```

Пользователю предлагается выбрать число от 0 до 4 - выбор блюда или получения чека. Если выбрано одно из 4 блюд, то выводится цена блюда, и она прибавляется к чеку. Важно отметить, что программа не завершится, пока пользователь не получит чек. (Ну либо, если он введёт не число, а какой-нибудь другой символ {не надо так делать :) }, то программа упадёт...)

Результат:

```
alexvasyukov@Alexanders-MacBook-Air 02 % chmod +x menu.sh
alexvasyukov@Alexanders-MacBook-Air 02 % ./menu.sh
Choose the dish:
1. Salad
2. Borsch
3. Pasta
4. Dessert
0. Check the bill.
Enter the number of option: 2
Borsch: 0.5 euro.

Choose the dish:
1. Salad
2. Borsch
3. Pasta
4. Dessert
0. Check the bill.
Enter the number of option: 1
Salad: 1.5 euro

Choose the dish:
1. Salad
2. Borsch
3. Pasta
4. Dessert
0. Check the bill.
Enter the number of option: 3
Pasta: 2 euro.

Choose the dish:
1. Salad
2. Borsch
3. Pasta
4. Dessert
0. Check the bill.
Enter the number of option: 4
Dessert: 1 euro.

Choose the dish:
1. Salad
2. Borsch
3. Pasta
4. Dessert
0. Check the bill.
Enter the number of option: 7
Use correct options. From 0 to 4.

Choose the dish:
1. Salad
2. Borsch
3. Pasta
```

```
4. Dessert
0. Check the bill.
Enter the number of option: 0
The bill is 5.0 euro.
```

5. backup_file.sh

   Проверяется существование файла, затем спрашивается, делать ли копию файла. Если введено "y" или "Y", то делается копия.

```
$ backup_file.sh
#!/bin/bash
# Backup of the file.

backup() {
    if [ ! -f "$1" ]; then
        echo "File '$1' not found."
        return 1
    fi
    timestamp=$(date +"%d%m%Y_%H%M%S")
    backup_name="${1}_backup_${timestamp}"
    cp "$1" "$backup_name"
    if [ $? -eq 0 ]; then
        echo "The backup of file '$1' is creating with name '$backup_name'."
    else
        echo "Error creating backup."
    fi
}

confirm() {
    read -p "Do you want to create the backup of file '$1'? (y/n): " answer
    if [ "$answer" = "y" ] || [ "$answer" = "Y" ]; then
        backup "$1"
    else
        echo "Backup canceled."
    fi
}

if [ -z "$1" ]; then
    echo "You need to enter the name of file"
    exit 1
fi

confirm "$1"
```

   Проверяется, что пользователь передал в параметре файл. Дальше спрашивается, делать ли копию. Если положительный ответ, то делается копия. В функции `backup` проверяется существование файла, потом формируется новое название файла из оригинального названия слова `backup` и даты, когда происходит копирование, и копируется.

Результат:

```
alexvasyukov@Alexanders-MacBook-Air 02 % chmod +x backup_file.sh
alexvasyukov@Alexanders-MacBook-Air 02 % ./backup_file.sh find_file.sh
Do you want to create the backup of file 'find_file.sh'? (y/n): n
Backup canceled.
alexvasyukov@Alexanders-MacBook-Air 02 % ./backup_file.sh find_file.sh
Do you want to create the backup of file 'find_file.sh'? (y/n): y
The backup of file 'find_file.sh' is creating with name 'find_file.sh_backup_02022025_221655'.
alexvasyukov@Alexanders-MacBook-Air 02 %
```

Появился файл `find_file.sh_backup_02022025_221655`: