

# Отчет по лабораторной работе №21 по курсу практикум на ЭВМ

Студент группы М8О-101Б-20 Ядров Артем Леонидович, № по списку 28

Контакты www, e-mail, icq, skype temayadrow@gmail.com

Работа выполнена: «    » \_\_\_\_\_ 202\_\_ г.

Преподаватель: доцент каф. 806 Никулин Сергей Петрович

Входной контроль знаний с оценкой \_\_\_\_\_

Отчет сдан «    » \_\_\_\_\_ 202\_\_ г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_

1. **Тема:** Программирование на интерпретируемых командных языках

2. **Цель работы:** Научиться программировать на интерпретируемом командном языке (bash)

3. **Задание (вариант № 28):** Интерпретация машины Тьюринга в четверках

4. **Оборудование (лабораторное):**  
ЭВМ Intel Pentium G2140, процессор 3.30 GHz, имя узла сети Cameron с ОП 8096 Мб,  
НМД 7906 Мб. Терминал ASUS адрес dev/pets/3 Принтер HP Laserjet 6P  
Другие устройства \_\_\_\_\_

*Оборудование ПЭВМ студента, если использовалось:*

Процессор Intel core i5-7300HQ 2.50 GHz с ОП 8096 Мб, НМД 131072 Мб. Монитор ASUS  
Другие устройства \_\_\_\_\_

5. **Программное обеспечение (лабораторное):**  
Операционная система семейства Unix, наименование Ubuntu версия 18.15.0  
интерпретатор команд bash версия 4.4.20  
Система программирования GNU версия 5.8.13  
Редактор текстов emacs версия 25.2.2  
Утилиты операционной системы cat  
Прикладные системы и программы \_\_\_\_\_  
Местонахождение и имена файлов программ и данных stud/208104

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства Unix, наименование Fedora версия 33  
интерпретатор команд bash версия 5.0.17  
Система программирования Clion версия 2020.3  
Редактор текстов emacs версия 25.2.2  
Утилиты операционной системы cat  
Прикладные системы и программы \_\_\_\_\_

Местонахождение и имена файлов программ и данных на домашнем компьютере home/Temi4

**6. Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Для начала проверим, вызван ли ключ «?». В случае вызова ключа выведем информацию о скрипте и закончим скрипт. В противном случае считаем параметры. В случае нехватки параметра запросим на ввод недостающие параметры. После прочтения проверим, являются ли введенные данные читаемыми файлами, выведем соответствующую информацию. Для удобной работы с лентой машины Тьюринга преобразуем строку из файла с тестом в массив. Установим индекс текущей позиции головки как переменную «pos», первоначально равную длине строки. Установим текущее состояние (в данном случае начальное) «q» равным 0 и текущий символ рабочей ячейки «a» равным пробелу. Запустим бесконечный цикл, выходом из которого могут служить 3 случая:

- В случае успешного завершения программы (для этого храним предыдущее состояние и предыдущий символ рабочей ячейки)
- В случае, когда конфигурации соответствует больше одной команды
- В случае, когда конфигурации не соответствует ни одной команды

С помощью команды «gтер» будем находить в файле с программой интересующую нас четверку.

Стоит заметить, что нас интересует лишь совпадение в начале строки («q,a» находится в начале строки)

В случае, если конфигурации соответствует одна команда, будем выводить эту команду, а затем выводить текущее состояние ленты.

**7. Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

**Тесты:**

1. ./turing.sh ?
2. ./turing.sh
3. ./turing.sh proga test
4. ./turing.sh test proga

Пункты 1-7 отчета составляются строго до начала лабораторной работы.

Допущен к выполнению работы. Подпись преподавателя \_\_\_\_\_

8. Распечатка протокола (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

```
[Temi4@localhost laabs]$ cat turing.sh
```

```
#!/bin/bash
```

```
for var; do
```

```
case $var in
```

```
'?')
```

```
echo "Скрипт интерпретирует МТ в 4"
```

```
echo "Использование: ./f1.sh [ФАЙЛ1] [ФАЙЛ2]"
```

```
echo -e "В первой строке ФАЙЛа1 должна быть записана начальная ситуация ленты.\nКонец исходных данных должен заканчиваться пробелом не должен быть загрязнен прочими символами"
```

```
echo "В ФАЙЛе2 должна быть записана программа без комментариев"
```

```
exit
```

```
;;
```

```
esac
```

```
done
```

```
ARGS=2
```

```
if [ $# -lt $ARGS ]; then
```

```
if [ $# -eq 0 ]; then
```

```
echo "Введите 1-й параметр"
```

```
read test
```

```
echo "Введите 2-й параметр"
```

```
read prog
```

```
else
```

```
echo "Введите параметр"
```

```
read prog
```

```
fi
```

```
elif [ $# -gt $ARGS ]; then
```

```
echo "Количество параметров превышает норму"
```

```
exit
```

```
else
```

```
test=$1
```

```
prog=$2
```

```
fi
```

```
if [ -r $test ] && [ -r $prog ]; then
```

```
echo "Параметры введены верно"
```

```
else
```

```
echo "Параметры введены неверно"
```

```
exit
```

```
fi
```

```
turing=()
```

```
str=$(head -n 1 $test)
```

```
for ((i = 0; i < ${#str}; i++)); do
```

```
turing[i]=$(echo "${str:$i:1}")
```

```
done
```

```
pos=${#turing[*]}
```

```
pos=$(( pos - 1))
```

```
a=' ' #Команда
```

```
s=' ' #Текущий символ
```

```
q='0' #Текущее состояние
```

```
while (true); do
```

```
n=$(grep "^$q,$s," $prog | wc -l)
```

```
if [ $n -gt 1 ]; then
```

```
echo "Для конфигурации q = $q, a = $s найдено больше 1 совпадения"
```

```
exit
```

```
elif [ 1 -gt $n ]; then
```

```
echo "Для конфигурации q = $q, a = $s не определена команда"
```

```
exit
```

```
else
```

```
str=$(grep "^$q,$s," $prog)
```

```
echo $str
```

```
a=$(echo "$str" | cut -d ',' -f 3)
```

```
q=$(echo "$str" | cut -d ',' -f 4)
```

```
q1=$(echo "$str" | cut -d ',' -f 1)
```

```
a1=$(echo "$str" | cut -d ',' -f 2)
```

```
if [[ "$a" = "1" ]]; then
```

```
pos=$((pos - 1))
```

```
if [ 0 -gt $pos ]; then
```

```

    echo "Программа вышла за левую границу ленты"
    exit
fi
s=${turing[$pos]}
elif [[ "$a" = "r" ]]; then
    pos=$(( $pos+1 ))
    if [ $pos -gt ${#turing[*]} ]; then
        turing[pos]=' '
    fi
    s=${turing[$pos]}
else
    turing[pos]=$a
    s=$a
fi
echo -n -e "\t\t"
for ((i = 0; i <= $pos; i++)); do
    echo -n ${turing[$i]}
done
echo "
if [[ "$q1" = "q" ]] && [[ "$a1" = "a" ]]; then
    echo "Программа успешно завершилась"
    for ((i = 0; i <= $pos; i++)); do
        echo -n ${turing[$i]}
    done
    echo "
    exit
fi
fi
done

```

[Temi4@localhost laabs]\$

[Temi4@localhost laabs]\$ cat test

aaaa

[Temi4@localhost laabs]\$ cat prog

0, ,1,0

0,a, ,1

1, ,1,1

1,a,b,1

1,b,c,1

1,c,d,1

1,d,d,1

[Temi4@localhost laabs]\$ ./turing.sh ?

Скрипт интерпретирует МТ в 4

Использование: ./f1.sh [ФАЙЛ1] [ФАЙЛ2]

В первой строке ФАЙЛа1 должна быть записана начальная ситуация ленты.

Конец исходных данных должен заканчиваться пробелом не должен быть загрязнен прочими символами

В ФАЙЛе2 должна быть записана программа без комментариев

[Temi4@localhost laabs]\$ ./turing.sh

Введите 1-й параметр

test

Введите 2-й параметр

prog

Параметры введены верно

0, ,1,0

aaaa

0,a, ,1

aaa

1, ,1,1

aaa

1,a,b,1

aab

1,b,c,1

aac

1,c,d,1

aad

1,d,d,1

aad

Программа успешно завершилась

aad

```

[Temi4@localhost laabs]$ cat > test
0
[Temi4@localhost laabs]$ cat +1
0, ,1,1
1,1,0,2
1,0,1,3
2,0,1,1
2,1,1,1
1, ,r,4
3,0,r,3
3,1,r,3
3, , ,3
4,0,1,5
5,1,r,5
5,0,r,5
5, ,0,6
6,0,r,6
6, , ,6
[Temi4@localhost laabs]$ ./turing.sh
Введите 1-й параметр
test
Введите 2-й параметр
+1
Параметры введены верно
0, ,1,1
0
1,0,1,3
1
3,1,r,3
1
3, , ,3
1
Программа успешно завершилась
1
[Temi4@localhost laabs]$ cat proga
0, ,1,1
1,1,0,5
1,0,1,2
2, ,r,3
3,0, ,3
3, , ,3
2,0,r,4
2,1,r,4
4,0,1,6
6,1,1,4
4,1,1,7
7, ,r,8
8,1, ,8
8, , ,8
7,0,r,9
7,1,r,9
9,0,0,5
9,1,0,5
5,1,r,5
5,0,r,5
5, , ,5
[Temi4@localhost laabs]$ ./turing.sh test proga
Параметры введены верно
0, ,1,1
0
1,0,1,2
2, ,r,3
0
3,0, ,3
3, , ,3

```

Программа успешно завершилась

```
[Temi4@localhost laabs]$ cat > test
```

1

```
[Temi4@localhost laabs]$ ./turing.sh test proga
```

Параметры введены верно

0, ,1

1

1,1,0,5

0

5,0,r,5

0

5, , ,5

0

Программа успешно завершилась

0

**9. Дневник отладки** должен содержать дату и время сеансов отладки, и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

**10. Замечания автора** по существу работы. Программу можно в дальнейшем улучшать, добавив возможность тестирования не одного теста, а **n** (в начале файла указывается **n**, а затем пишется **n** строк теста). Также существует возможность добавления разнообразных ключей (отключения вывода текущей ситуации, вывода текущей команды, записи вывода в файл)

#### 11. Выводы

Я изучил интерпретируемый командный язык (bash) и научился писать на нем программы.

Недочёты при выполнении задания могут быть устранены следующим образом:

Подпись студента \_\_\_\_\_