Отчёт по лабораторной работе № 7 по курсу «Архитектура компьютера и информационных систем»

Студент группы М8О-112Б-22

Епифанов Евгений Валерьевич, № по списку: 9

# E-mail: [epi\_abitur](mailto:epi_abitur)@mail.ru

# Телефон: 8(916)783-95-69

# Работа выполнена: 5 октября 2022 г.

Преподаватель: Никулин Сергей Петрович, каф.806

**1. Тема:** программирование в алгоритмической модели Маркова.

**2. Цель работы:** научиться работать с алгоритмической моделью Маркова и использовать её в терминале Linux.

**3. Задание:** изучить синтаксис команд алгоритма Маркова, составить программу, решающую полученную задачу (7 вариант: Входное слово представляет собой два троичных числа без знака, разделенные знаком “<”. Составить алгоритм вычисления троичного логического сдвига первого числа влево на число разрядов второго числа), и продемонстрировать результат работы в виде протокола.

**4. Оборудование (ПЭВМ студента):**

Ноутбук с процессором Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz, 1992 МГц, ядер: 4, логических процессоров: 8, ОП 8 ГБ. SSD 256 ГБ.

**5. Программное обеспечение ЭВМ студента:**

* Операционная система семейства UNIX, наименование: Linux Ubuntu (22.04) 5.15.0-48-generic #54-Ubuntu SMP Fri Aug 26 13:26:29 UTC 2022 x86\_64 x86\_64 x86\_64 GNU/Linux.
* Интерпретатор команд: GNU bash, version 5.1.16(1)-release (x86\_64-pc-linux-gnu).
* Редактор текстов: GNU Emacs 28.2.
* Утилиты и прочие программы.
* nam (Linux)

**6. Идея, метод, алгоритм решения задачи:**

1. Синтаксис команд алгоритма Маркова.

1.1 – -> – замена символов.

1.2 – ->. – замена символов с последующим завершением программы.

1. Запись программы и тестов в файл.

2.1 cat > test.nam – запись программы в файл.

1. Отображение решения задачи.

3.1 ./nam – запуск пошагового отображения работы программы.

1. Описание логического сдвига.

4.1. При логическом сдвиге влево происходит сдвиг всех разрядов регистра влево на одну позицию, старший бит (самый левый) при этом теряется, а в младший (самый правый) записывается 0.

**7. Сценарий выполнения работы:**

1. Изучить синтаксис команд для работы с алгоритмической моделью Маркова

и написать программу для решения полученной задачи.

1. Приготовить тесты для проверки работоспособности программы.

Ввод: 1<1

Вывод: 0

Ввод: 210<12

Вывод: 000

Ввод: 1102201<102

Вывод: 2201000

Ввод: 2112<20

Вывод: 1200

Ввод: 10002<111

Вывод: 02000

1. Использовать команды терминала Linux для демонстрации

работоспособности написанной программы. Проверить программу на заранее заготовленных тестах.

**8. Распечатка протокола:**

evgeniy2@evgeniy2:~$ cat > test.nam

)0->0)

)1->1)

)2->2)

)<-><)

)->(

2<((->((20<

1<((->((10<

0<((->((00<

<(->.

0((->((0

1((->((1

2((->((2

0(->((

1(->((

2(->((

((1->)

((2->)

((0->)

<-><)

^C

evgeniy2@evgeniy2:~$ ./nam test.nam

^C

Enter new data or press CTRL/C to exit

)0 -> 0)

)1 -> 1)

)2 -> 2)

)< -> <)

) -> (

2<(( -> ((20<

1<(( -> ((10<

0<(( -> ((00<

<( ->.

0(( -> ((0

1(( -> ((1

2(( -> ((2

0( -> ((

1( -> ((

2( -> ((

((1 -> )

((2 -> )

((0 -> )

< -> <)

evgeniy2@evgeniy2:~$ cat nam.out

nam: test.nam of Mon Oct 17 20:34:30 2022

)0 -> 0)

)1 -> 1)

)2 -> 2)

)< -> <)

) -> (

2<(( -> ((20<

1<(( -> ((10<

0<(( -> ((00<

<( ->.

0(( -> ((0

1(( -> ((1

2(( -> ((2

0( -> ((

1( -> ((

2( -> ((

((1 -> )

((2 -> )

((0 -> )

< -> <)

1<1

--------------------------------------------------------------------------------

[ < -> <) ] "1<)1"

[ )1 -> 1) ] "1<1)"

[ ) -> ( ] "1<1("

[ 1( -> (( ] "1<(("

[ 1<(( -> ((10< ] "((10<"

[ ((1 -> ) ] ")0<"

[ )0 -> 0) ] "0)<"

[ )< -> <) ] "0<)"

[ ) -> ( ] "0<("

[ <( ->. ] "0"

--------------------------------------------------------------------------------

210<12

--------------------------------------------------------------------------------

[ < -> <) ] "210<)12"

[ )1 -> 1) ] "210<1)2"

[ )2 -> 2) ] "210<12)"

[ ) -> ( ] "210<12("

[ 2( -> (( ] "210<1(("

[ 1(( -> ((1 ] "210<((1"

[ 0<(( -> ((00< ] "21((00<1"

[ 1(( -> ((1 ] "2((100<1"

[ 2(( -> ((2 ] "((2100<1"

[ ((2 -> ) ] ")100<1"

[ )1 -> 1) ] "1)00<1"

[ )0 -> 0) ] "10)0<1"

[ )0 -> 0) ] "100)<1"

[ )< -> <) ] "100<)1"

[ )1 -> 1) ] "100<1)"

[ ) -> ( ] "100<1("

[ 1( -> (( ] "100<(("

[ 0<(( -> ((00< ] "10((00<"

[ 0(( -> ((0 ] "1((000<"

[ 1(( -> ((1 ] "((1000<"

[ ((1 -> ) ] ")000<"

[ )0 -> 0) ] "0)00<"

[ )0 -> 0) ] "00)0<"

[ )0 -> 0) ] "000)<"

[ )< -> <) ] "000<)"

[ ) -> ( ] "000<("

[ <( ->. ] "000"

--------------------------------------------------------------------------------

1102201<102

--------------------------------------------------------------------------------

[ < -> <) ] "1102201<)102"

[ )1 -> 1) ] "1102201<1)02"

[ )0 -> 0) ] "1102201<10)2"

[ )2 -> 2) ] "1102201<102)"

[ ) -> ( ] "1102201<102("

[ 2( -> (( ] "1102201<10(("

[ 0(( -> ((0 ] "1102201<1((0"

[ 1(( -> ((1 ] "1102201<((10"

[ 1<(( -> ((10< ] "110220((10<10"

[ 0(( -> ((0 ] "11022((010<10"

[ 2(( -> ((2 ] "1102((2010<10"

[ 2(( -> ((2 ] "110((22010<10"

[ 0(( -> ((0 ] "11((022010<10"

[ 1(( -> ((1 ] "1((1022010<10"

[ 1(( -> ((1 ] "((11022010<10"

[ ((1 -> ) ] ")1022010<10"

[ )1 -> 1) ] "1)022010<10"

[ )0 -> 0) ] "10)22010<10"

[ )2 -> 2) ] "102)2010<10"

[ )2 -> 2) ] "1022)010<10"

[ )0 -> 0) ] "10220)10<10"

[ )1 -> 1) ] "102201)0<10"

[ )0 -> 0) ] "1022010)<10"

[ )< -> <) ] "1022010<)10"

[ )1 -> 1) ] "1022010<1)0"

[ )0 -> 0) ] "1022010<10)"

[ ) -> ( ] "1022010<10("

[ 0( -> (( ] "1022010<1(("

[ 1(( -> ((1 ] "1022010<((1"

[ 0<(( -> ((00< ] "102201((00<1"

[ 1(( -> ((1 ] "10220((100<1"

[ 0(( -> ((0 ] "1022((0100<1"

[ 2(( -> ((2 ] "102((20100<1"

[ 2(( -> ((2 ] "10((220100<1"

[ 0(( -> ((0 ] "1((0220100<1"

[ 1(( -> ((1 ] "((10220100<1"

[ ((1 -> ) ] ")0220100<1"

[ )0 -> 0) ] "0)220100<1"

[ )2 -> 2) ] "02)20100<1"

[ )2 -> 2) ] "022)0100<1"

[ )0 -> 0) ] "0220)100<1"

[ )1 -> 1) ] "02201)00<1"

[ )0 -> 0) ] "022010)0<1"

[ )0 -> 0) ] "0220100)<1"

[ )< -> <) ] "0220100<)1"

[ )1 -> 1) ] "0220100<1)"

[ ) -> ( ] "0220100<1("

[ 1( -> (( ] "0220100<(("

[ 0<(( -> ((00< ] "022010((00<"

[ 0(( -> ((0 ] "02201((000<"

[ 1(( -> ((1 ] "0220((1000<"

[ 0(( -> ((0 ] "022((01000<"

[ 2(( -> ((2 ] "02((201000<"

[ 2(( -> ((2 ] "0((2201000<"

[ 0(( -> ((0 ] "((02201000<"

[ ((0 -> ) ] ")2201000<"

[ )2 -> 2) ] "2)201000<"

[ )2 -> 2) ] "22)01000<"

[ )0 -> 0) ] "220)1000<"

[ )1 -> 1) ] "2201)000<"

[ )0 -> 0) ] "22010)00<"

[ )0 -> 0) ] "220100)0<"

[ )0 -> 0) ] "2201000)<"

[ )< -> <) ] "2201000<)"

[ ) -> ( ] "2201000<("

[ <( ->. ] "2201000"

--------------------------------------------------------------------------------

2112<20

--------------------------------------------------------------------------------

[ < -> <) ] "2112<)20"

[ )2 -> 2) ] "2112<2)0"

[ )0 -> 0) ] "2112<20)"

[ ) -> ( ] "2112<20("

[ 0( -> (( ] "2112<2(("

[ 2(( -> ((2 ] "2112<((2"

[ 2<(( -> ((20< ] "211((20<2"

[ 1(( -> ((1 ] "21((120<2"

[ 1(( -> ((1 ] "2((1120<2"

[ 2(( -> ((2 ] "((21120<2"

[ ((2 -> ) ] ")1120<2"

[ )1 -> 1) ] "1)120<2"

[ )1 -> 1) ] "11)20<2"

[ )2 -> 2) ] "112)0<2"

[ )0 -> 0) ] "1120)<2"

[ )< -> <) ] "1120<)2"

[ )2 -> 2) ] "1120<2)"

[ ) -> ( ] "1120<2("

[ 2( -> (( ] "1120<(("

[ 0<(( -> ((00< ] "112((00<"

[ 2(( -> ((2 ] "11((200<"

[ 1(( -> ((1 ] "1((1200<"

[ 1(( -> ((1 ] "((11200<"

[ ((1 -> ) ] ")1200<"

[ )1 -> 1) ] "1)200<"

[ )2 -> 2) ] "12)00<"

[ )0 -> 0) ] "120)0<"

[ )0 -> 0) ] "1200)<"

[ )< -> <) ] "1200<)"

[ ) -> ( ] "1200<("

[ <( ->. ] "1200"

--------------------------------------------------------------------------------

10002<111

--------------------------------------------------------------------------------

[ < -> <) ] "10002<)111"

[ )1 -> 1) ] "10002<1)11"

[ )1 -> 1) ] "10002<11)1"

[ )1 -> 1) ] "10002<111)"

[ ) -> ( ] "10002<111("

[ 1( -> (( ] "10002<11(("

[ 1(( -> ((1 ] "10002<1((1"

[ 1(( -> ((1 ] "10002<((11"

[ 2<(( -> ((20< ] "1000((20<11"

[ 0(( -> ((0 ] "100((020<11"

[ 0(( -> ((0 ] "10((0020<11"

[ 0(( -> ((0 ] "1((00020<11"

[ 1(( -> ((1 ] "((100020<11"

[ ((1 -> ) ] ")00020<11"

[ )0 -> 0) ] "0)0020<11"

[ )0 -> 0) ] "00)020<11"

[ )0 -> 0) ] "000)20<11"

[ )2 -> 2) ] "0002)0<11"

[ )0 -> 0) ] "00020)<11"

[ )< -> <) ] "00020<)11"

[ )1 -> 1) ] "00020<1)1"

[ )1 -> 1) ] "00020<11)"

[ ) -> ( ] "00020<11("

[ 1( -> (( ] "00020<1(("

[ 1(( -> ((1 ] "00020<((1"

[ 0<(( -> ((00< ] "0002((00<1"

[ 2(( -> ((2 ] "000((200<1"

[ 0(( -> ((0 ] "00((0200<1"

[ 0(( -> ((0 ] "0((00200<1"

[ 0(( -> ((0 ] "((000200<1"

[ ((0 -> ) ] ")00200<1"

[ )0 -> 0) ] "0)0200<1"

[ )0 -> 0) ] "00)200<1"

[ )2 -> 2) ] "002)00<1"

[ )0 -> 0) ] "0020)0<1"

[ )0 -> 0) ] "00200)<1"

[ )< -> <) ] "00200<)1"

[ )1 -> 1) ] "00200<1)"

[ ) -> ( ] "00200<1("

[ 1( -> (( ] "00200<(("

[ 0<(( -> ((00< ] "0020((00<"

[ 0(( -> ((0 ] "002((000<"

[ 2(( -> ((2 ] "00((2000<"

[ 0(( -> ((0 ] "0((02000<"

[ 0(( -> ((0 ] "((002000<"

[ ((0 -> ) ] ")02000<"

[ )0 -> 0) ] "0)2000<"

[ )2 -> 2) ] "02)000<"

[ )0 -> 0) ] "020)00<"

[ )0 -> 0) ] "0200)0<"

[ )0 -> 0) ] "02000)<"

[ )< -> <) ] "02000<)"

[ ) -> ( ] "02000<("

[ <( ->. ] "02000"

--------------------------------------------------------------------------------

9. Дневник отладки:

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

10. Замечания по существу работы: -

11. Выводы: Я научился составлять программы для алгоритмической модели Маркова и демонстрировать их работу на различных тестах.

Недочёты при выполнении задания могут быть устранены следующим образом: -