

Задание: «Моделирование архитектуры фон Неймана на языке программирования Си»

Цель: понять ключевые принципы архитектуры фон Неймана — единое хранилище для данных и команд, последовательное выполнение, наличие счётчика команд — через написание простой программы на Си без использования указателей, структур.

Общие требования:

1. Объявите массив типа **int**, который будет хранить вашу программу и заполните его вручную (в коде) командами.
2. Используйте переменную **int** для организации счётчика команд.
3. Используйте переменную **int** которая моделирует «регистр», где хранится текущий результат выполнения вашей программы.
4. Программа должна работать в цикле **while**, пока не встретит команду 0 (это будет «стоп»).
5. Выводите на экран каждую выполняемую команду и значение ее значение в моменте после её выполнения.
6. Используйте только конструкции, которые вы уже изучили.

Уровень «Удовлетворительно»

Варианты программ (выберите один):

0,5	1 — прибавить число к хранимому в регистре 2 — вычесть число из хранимого в регистре 3 — уменьшить значение, хранимое в регистре на 1 0 — стоп	После кода команды (1 или 2) в массиве сразу идёт число-операнд. Команда 3 не требует операнд.
1,6	1 — умножить на хранимое в регистре число 2 — разделить на хранимое в регистре число 3 — увеличить значение, хранимое в регистре на 1 0 — стоп	После кода команды (1 или 2) в массиве сразу идёт число-операнд. Команда 3 не требует операнд.

2,7	1— загрузить число хранимое в регистре 2— прибавить к хранимому в регистре числу 3 — вывести значение, хранимое в регистре 0— стоп	После кода команды (1 или 2) в массиве сразу идёт число-операнд. Команда 3 не требует операнд.
3,8	1— увеличить значение, хранимое в регистре на 1 2— уменьшить значение, хранимое в регистре на 1 3— прибавить число 0— стоп	Команды 1 и 2 не требуют операнда. Команда 3 — требует (слева дующая ячейка). После кода команды 3 в массиве сразу идёт число-операнд.
4,9	1 — вывести значение, хранимое в регистре 2 — загрузить число 3— умножить на хранимое в регистре число 0 — стоп	Команда 1 просто печатает текущее значение (без изменения). Команда 2 загружает число из следующей ячейки. После кода команды 3 в массиве сразу идёт число-операнд.

Пример (для варианта 1):

Входной массив: {1, 10, 2, 4, 1, 3, 0}

Выполнение:

- $pc=0$: команда 1 \rightarrow взять $program[1]=10 \rightarrow acc = 0 + 10 = 10 \rightarrow pc += 2$
- $pc=2$: команда 2 \rightarrow взять $program[3]=4 \rightarrow acc = 10 - 4 = 6 \rightarrow pc += 2$
- $pc=4$: команда 1 \rightarrow взять $program[5]=3 \rightarrow acc = 6 + 3 = 9 \rightarrow pc += 2$
- $pc=6$: команда 0 \rightarrow стоп

Вывод программы:

Выполняется команда 1, $acc = 10$

Выполняется команда 2, $acc = 6$

Выполняется команда 1, $acc = 9$

Программа завершена.

Уровень «хорошо»

Требования уровня 3 + следующее:

Добавьте в вашу систему команду 4 — безусловный переход (JUMP).

- Команда 4 означает: «перейти на указанную ячейку памяти».
- После команды 4 в массиве сразу идёт адрес перехода (целое число — индекс в массиве, где хранится ваша программа).
- После выполнения команды 4 значение счётчика команд заменяется на этот адрес (а не увеличивается как обычно).

Пример:

Массив: {1, 5, 4, 6, 2, 1, 0}

Выполнение:

- счётчика команд 0: команда 1 → массив который хранит вашу программу = 5, счётчика команд = 2
- счётчика команд 2: команда 3 → перейти на ячейку памяти = 6 → счётчика команд = 6
- счётчика команд 6: команда 0 → стоп
Команды 2, 1 никогда не выполняются.

Важно:

- Убедитесь, что переход не уходит за пределы массива.
- После команды 3 не увеличивайте счётчика команд на 2, а присвойте ему значение из следующей ячейки.

Уровень «отлично»

Требования уровня 4 + следующее:

Добавьте команду 5 — условный переход (JUMP IF ZERO).

- Команда 5 означает: «если значение регистра == 0, перейти на указанный адрес; иначе — продолжить как обычно».
- После команды 5 в массиве идёт адрес перехода.
- Если значение регистра != 0, то команда игнорируется, и значение счётчика команд увеличивается на 2 (как для обычных команд с операндом).

Пример:

Массив: {1, 0, 4, 8, 1, 99, 0, 0, 1, 42, 0}

Выполнение:

- значение счётчика команд = 0: команда 1 → в массив, который хранит вашу программ = 0, значение счётчика команд = 2
 - значение счётчика команд = 2: команда 4 → в массив который хранит вашу программ = 0 → перейти на `program[3] = 8` → значение счётчика команд = 8
 - значение счётчика команд = 8: команда 1 → в массив, который хранит вашу программ = 42, значение счётчика команд = 10
 - значение счётчика команд = 10: команда 0 → стоп
- Команды 1, 99, 0, 0 пропущены.

Требования к оформлению отчёта

Отчёт оформляется в электронной форме и должен содержать следующие разделы:

1. Титульный лист

- По шаблону СибГУТИ
- Название работы: «Моделирование архитектуры фон Неймана на языке C»
- ФИО студента, группа
- Вариант задания (указывается номер: 0–9)

2. Цель работы

Кратко (2–3 предложения):

Пример: Понять ключевые принципы архитектуры фон Неймана — единое хранилище для данных и команд, последовательное выполнение, наличие счётчика команд — через написание простой программы на языке C.

3. Описание программы

- Указать, какой уровень реализован: «3», «4» или «5».
- Описать логику работы программы:
- Какие команды поддерживает ваша «виртуальная машина»?
- Как организован массив-«память»?
- Как работает счётчик команд?
- Какие особенности реализованы (например, безусловный или условный переход)?
- Привести пример массива программы и пошагово объяснить её выполнение (аналогично примеру из задания).

4. Листинг программы

- Полный исходный код на языке C.
- Код должен быть читаемым: с отступами, комментариями и осмысленными именами переменных.
- Обязательно включить комментарий, поясняющий, как программа отражает принципы архитектуры фон Неймана.

5. Результаты выполнения

- Привести реальный вывод программы (скриншот или текст).
- Указать, какие входные данные использовались (если есть ввод).

- Для уровней «4» и «5» — показать, как работает **JUMP** или условный переход (например, цикл или ветвление).

6. Вывод

Краткое заключение (3–4 предложения):

- Что удалось реализовать?
- Какие принципы архитектуры фон Неймана были продемонстрированы?
- Были ли трудности? Как они были решены?

Важно:

- Отчёт должен быть индивидуальным. Использование шаблонных фраз или копирование чужого кода без понимания не допускается.