

Лабораторная работа 4.

Выполнение команд процесса

Цель.

Изучить выполнение операций команды активного процесса центрального процессора на основе Фон-Неймановской архитектуры ЭВМ.

Теоретическая часть.

Последовательность операций для выполнения одной команды активного процесса на центральном процессоре строить на основе Фон-Неймановской структуры ЭВМ.

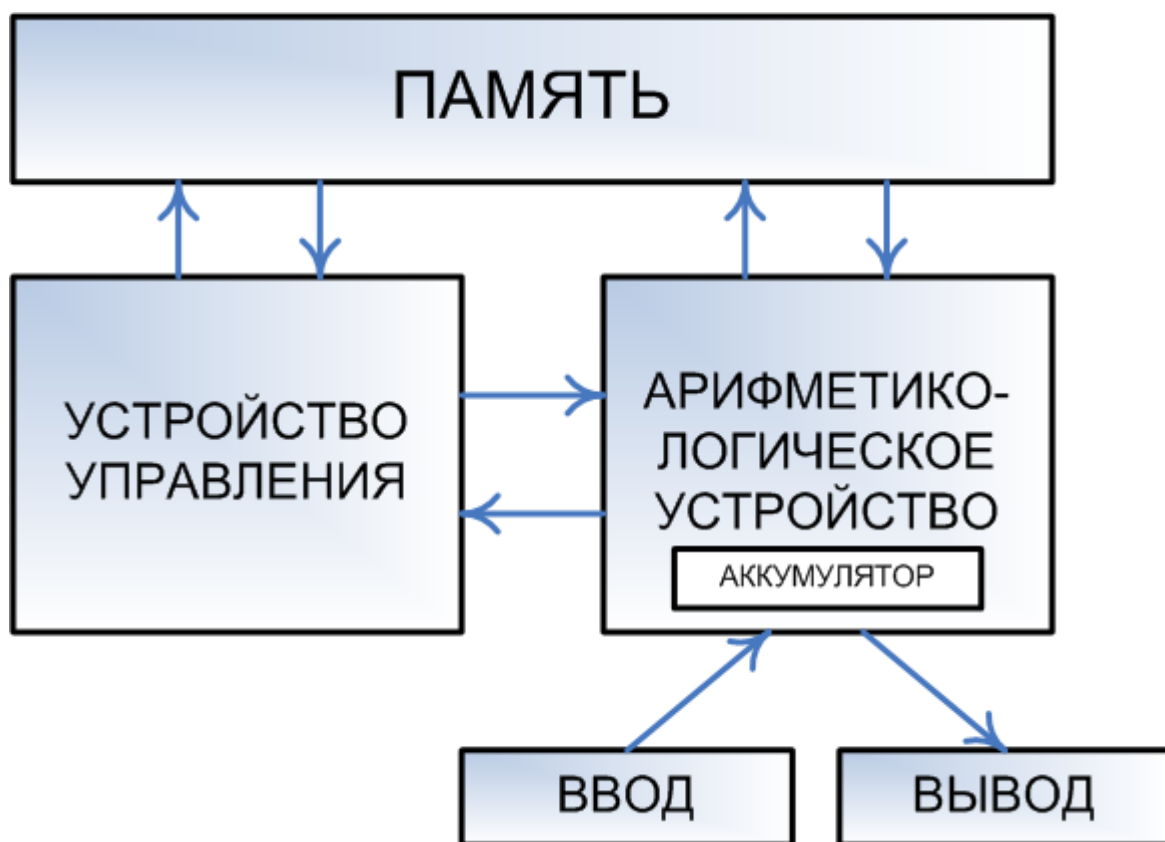


Рис. Схематичное изображение машины фон Неймана

Согласно фон Нейману, ЭВМ состоит из следующих основных блоков:

- 1) устройства ввода/вывода информации;
- 2) памяти ЭВМ;
- 3) процессора, включающего устройство управления (УУ) и арифметико-логическое устройство (АЛУ)

В ходе работы ЭВМ информация через устройства ввода попадает в память. Процессор извлекает из памяти обрабатываемую информацию, работает с ней и помещает в нее результаты обработки. Полученные результаты через устройства вывода передаются пользователю.

Алгоритм работы модуля моделирующего работу с памятью.

1. Чтение команды из памяти по адресу на аппаратном счетчике команд. В модели ОС выполняется как генерация команды Command в соответствии с процентным числом команд ввода (вывода) и временем выполнения задания. Эта подпрограмма определяет длительность задания путем генерации команды завершения (Exit, Quit, End), если счетчик команд равен числу команд задания. Для контроля можно предусмотреть генерацию ошибочных команд.
2. Увеличение аппаратного счетчика команд на длину команды. Для модели ОС длина команды = 1.
3. Распознавание команды (подпрограмма Get_CodOperation_Adress1_Adress2).
4. Чтение операндов из аппаратной памяти.
5. Выполнение операции CodOperation (подпрограмма Do_Operation).
6. Запись результата операции в аппаратную память.

Последовательность команд процесса генерируется случайным образом из набора:

- вычислительные команды, выполняемые на ЦПр за один такт моделируемого времени,
- команды обращения к устройствам ввода (вывода), выполняемые за N_InOut тактов,
- команда завершения задания, выполняется за время выгрузки задания, или за 1 такт.

Алгоритм работы модуля моделирующего работу с арифметико-логического устройства.

Для выполнения операций создайте подпрограмму Do_Operation моделирующую работу арифметико-логического устройства при выполнении операций:

- операция вычислительная: выполняется любая двухместная операция над операндами,
- операция ввода (вывода): порождается системный процесс инициализации процессора ввода (вывода), сохраняется слово состояния процесса, затем процесс переводится в состояние «Блокирован по обращению ко вводу (выводу)»,
- операция завершения задания: порождается системный процесс завершения задания (например, устанавливается отсутствие этого процесса или удаляется запись о процессе из очереди готовности, освобождается память, занимаемая данными этого процесса (динамическая память) и самим процессом (статическая память) из списка занятой памяти и освободившаяся память добавляется в список свободной памяти, уменьшается количество резидентных заданий, увеличивается число выполненных заданий, увеличивается количество свободных записей в таблице слов состояний процессов и т.д.). После выгрузки процесса можно загрузить новое задание, если есть достаточная память, или загрузка процесса выполняется и при условии, что центральный процессор простаивает (список готовности пуст).

Практическая часть

1. Добавить состояние «Блокирован по обращению ко вводу (выводу)». Список возможных состояний: «Готов», «Активен», «Инициализация IO», «Блокирован».
2. Реализовать подпрограмму «Выполнение команды процесса» с использованием следующих подпрограмм:
 - «Генерация команды процесса»,
 - «Разборка команды и получение кода операции»,
 - «Чтение операнда из памяти»,
 - «Выполнение операции»,
 - «Инициализация процессора ввода (вывода)»,
 - «Завершение процесса» (с загрузкой одного или нескольких новых заданий),

- «Запись результата операции в память».

3. Изменить подпрограмму индикации, добавив вывод типа выполняемой команды.

В отчете представить:

1. Блок схемы алгоритмов
2. Анализ программного кода
3. Листинг программы
4. Экранные формы