Знакомство

История вычислительных машин

Первая вычислительная машина была создана Чарльзом Беббиджем в начале 19 века. Она была механической.

Следующим человеком был Карл Цузи. Его вычислительные машины (Z1, Z2, … Z22) были релейными и работали в двоичной системе счисления.

Затем над созданием вычислительных машин уже трудились группы ученых. Среди самых известных – это Эниак (США), МЭСМ, БЭСМ (СССР). Которые уже являлись электронными вычислительными машинами.

Поколения ЭВМ.

Ламповые. Лампа представляет из себя колбу с вакуумом.

Транзисторные.

Интегральные схемы. собрана логическая структура.

Пример: на вход интегральной схеме подаются два сигнала. На выходе получается сигнал (т.е. напряжение сигнала увеличивается на конкретное значение. Для логической операции «И» на выходе появится цифровой сигнал, только если на вход были одновременно были поданы 2 сигнала. Для логического «ИЛИ» будет достаточно появление 1 сигнала.

СверхБольшиеИнтегральныеСхемы.

Эти схемы представляют собой микропроцессор или микроконтроллер. Однако у последних имеются отличия. Так, микропроцессор работает на архитектуре фон Неймана, а микроконтроллер на гарвардской архитектуре.

Принципы фон Неймана.

1. Принцип двоичности.
2. Принцип программного управления.
3. Принцип однородности памяти.

Необходим процессор (CPU) и оперативное запоминающее устройство (RAM). И данные и команды хранятся в одном месте. Преимущество такой концепции состоит в том, что программа может быть загружена в оперативную память и начать выполняться.

В гарвардской архитектуре существует память с запрещенной записью.

1. Принцип адресуемости памяти.

Это означает, что у каждой ячейки памяти есть свой номер.

1. Принцип последовательного программного управления.

Впрочем линейные программы не всегда удобны. Часто необходимы циклы. Однако безусловные переходы могут зациклить программу. Для устранения этой трудности необходим условный переход, в котором условие и последующие команды обеспечивают выход из цикла.

Так, например, пусть в условие заключается в сравнении с переменной с некоторым числом. Если переменная больше заданного числа, то программа выполняется по сценарию А. При этом переменная уменьшается на единицу и в какой-то момент станет меньше или равна заданной величины. Тогда выполнится сценарий В.

Замечательно то, что циклов и условных переходов достаточно для реализации любого алгоритма. Потому последним принципом фон Неймана является

1. Принцип условного перехода.

0034

Организация команд

Рассмотрим центральное процессорное устройство (ЦПУ). Также есть память (ОЗУ). Процессор представляет собой арифметико-логическое устройство, в котором реализованы сумматоры, мультипликаторы и другие логические операторы. Также ЦПУ содержит устройство управления (УУ), которое знает язык машинных команд.

Например, …

Для работы т=процессору необходима локальная память, которая называется регистром.

Разговор о шеснадцатиричной системе счисления. 5:20

Регистры имеют имена, такие как AX, BX, CX, DX, EX, которые являются арифметическими регистрами, т.е. в них записываются числа. Есть регистр IP, не имеющий ничего общего с сетью Интернет, в котором хранится адрес текущей выполняемой команды. (исполняемой программы).

Доступ к этой части памяти осуществляется через провод. При этом один провод за один такт работы компьютера может передать всего 1 бит информации. Поэтому для ускорения процесса используется 8 проводов, которые вместе называются шиной данных.

Дополнительно необходимо осуществлять не только запись но и считывание информации.

6:00

Необходимо управлять процессами записи в память, считывания из нее, а также доступа к другим устройствам. Этот доступ осуществляется посредством обращения к микроконтроллеру, в котором заложены программы… Для управлениями процессами считывания и записи данных необходима шина управления.

Для записи информации в память компьютера нужно передать сначала адрес ячейки памяти, а потом уже саму информацию. Чтобы сделать это одновременно используется дополнительная шина адресов.

Эти три шины вместе называют магистралью.

Допустим, процессор занят вычислениями, например, раскладывает число на множители. В некоторый момент пользователь вызывает некую команду, например нажимает CTRL-C. Для того чтобы ее обработать устройство вызывает аппаратное прерывание посредством обращения к контроллеру прерываний.

Во время прерывания регистры процессора перезаписываются, т.к. вместо их старых значений нужны данные для обработки другого процесса. Для того чтобы по окончании прерывания вернуть регистрам первоначальные значения, перед обработкой команд прерывания, значения регистров складываются в специальный сегмент памяти.

Для этого существует специальный регистр stack-pointer (SP), который указывает (?кому?) на адрес этого участка (?а если участок не достаточного размера?). Первым записывается значение IP, содержащее адрес текущей программы: в противном случае будет невозможно вернуться к ее выполнению. Затем последовательно загружаются значения других регистров.

(? Адрес последнего элемента записывается в SP?)

(? Как осуществляется возврат значений?)

После этого процессору нужно определить, в каком участке памяти располагаются команды обработчика прерывания. Для этого он обращается к началу оперативной памяти – таблице прерываний – содержащей адреса всех таких обработчиков, и по номеру прерывания получает соответствующий адрес.

связка

Файловая система.

Жесткий диск представляет собой стопку из нескольких дисков, каждый из которых разбит на кольца и сектора. При этом адрес ячейки памяти состоит из номера поверхности диска (head), номера кольца () и сектора (). Однако для удобства программирования и хранения файлов данные должны быть удобно структурированы, т.е. должна существовать файловая система. Ее организацией и расшифровкой значения адреса по имени файла, например,

/home/student/1.txt

занимается операционная система. Более того возникла необходимость в создании разделов жесткого диска. Расшифровать!!! Для доступа к ним в памяти есть специальная таблица разделов. Для работы с памятью такого виртуального диска, вызывается программное прерывание. При этом в регистры записывается задача: в AX – адрес начала имени файла, в BX – права доступа к этому файлу, и т.п. После этого вызывается стандартная процедура , которая и осуществляет работу с жестким диском. Эти стандартные обработчики прерываний записываются в оперативную память при загрузке BIOS и операционной системы. Это позволяет программному обеспечению, вызывая команду INT 0x21, запустить операционную систему, которая определяет задачу по значениям регистров и выполняет ее.

Иерархия – ОС

Изучив машинный язык или ассемблер для соответствующей архитектуры процессора можно написать ПО (software), оперирующее, например, непосредственно с жестким диском и ОЗУ (hardware). Однако это весьма трудоемкий процесс для программиста.

Операционная система занимает промежуточное место между программами и машинными командами. Таким образом, программа (application) может работать с регистрами и оперативной памятью. Команды же связанные с устройством (чего?) выполняются только ОС.

Начиная с 386 компьютера архитектуры Intel, появился защищенный режим работы компьютера. В нем процессор с помощью контроллера памяти определяет, обладает ли тот сегмент памяти, из которого берутся команды, привилегиями операционной системы. Если это так, программа(?) может выполнять запрещенные для приложения ассемблерные команды и, ей доступна прямая физическая адресация памяти. (определение?)

Приложение хранится в виртуальной памяти. Существует специальный контроллер памяти (тот же, может, виртуальной?), обычно встроенный в процессор, который содержит таблицу соответствий сегментов этой памяти сегментам физической памяти компьютера.

При записи приложением некоторого числа по адресу $7531 контроллер памяти определяет, что данный адрес соответствует, например, $751B.

Этой таблицей соответствий управляет операционная система, тем самым распределяя ресурсы компьютера.

Виртуальная машина.

Работая на одном компьютере можно использовать разные операционные системы. Для этого необходимо, чтобы в главной операционной системе в качестве приложения работала виртуальная машина – программа, эмулирующая работу компьютера и запускающая другую ОС.

Компиляция и интерпретация в высокоуровневых и низкоуровневых языках программирования.

Писать ассемблерный код достаточно трудоемко, поэтому этот язык используют только для оптимизации частей программы. Приложения пишут на более удобных языках программирования. После написания кода программы в итоге нужно получить бинарный исполняемый файл, который специфичен для каждой архитектуры. Этот перевод в некоторое количество этапов осуществляется специальными программами – компиляторами. Чтобы отложить этот процесс и тем самым позволить программе запускаться на любой архитектуре, необходим интерпретатор ( как он работает? его плюсы и минусы). Python является одним из интерпретируемых языков программирования.

4 свободы программного обеспечения.

Ричардом Столманом, основателем проекта GNU, были сформулированы следующие свободы программного обеспечения:

1. Cвобода использования

Это значит, что ПО может бесплатно использоваться кем угодно, когда, как и для чего угодно.

1. Свобода видеть исходный код программы (open source).
2. Свобода копирования и распространения.

Встречаются программы, включающие такую свободу, однако в них не предусмотрен доступ к исходному коду (shareware).

1. Свобода соучастия в разработке.

№№№33\_3:33