

Список вопросов, необходимо выучить

23.05.25 первые 5 вопросов так и идём

- 1 Общая структура вычислительной машины. Шины, процессоры, оперативная память, внешние устройства.
- 2 Понятие архитектуры процессора и микроархитектуры. Основные компоненты.
- 3 Принципы архитектуры Джона фон Неймана. Цикл выполнения команды. минимально необходимый набор регистров.
- 4 Первые ЭВМ, первые устройства ввода и вывода. Учебные машины УМ-3, УМ-2, УМ-1, как примеры организации первых ЭВМ.
- 5 Машинное представление целых чисел: прямой код, дополнительный код. Выставление флагов знакового и беззнакового переполнения. Флаг знака и флаг нуля. Диапазоны знаковых и беззнаковых чисел. Получение обратного числа через побитовое отрицание и сложение.

Список вопросов по курсу «Архитектура ЭВМ и язык ассемблера»

1. Общая структура вычислительной машины. Шины, процессоры, оперативная память, внешние устройства.
2. Понятие архитектуры процессора и микроархитектуры. Основные компоненты.
3. Принципы архитектуры Джона фон Неймана. Цикл выполнения команды. минимально необходимый набор регистров. Число операндов в команде. CISC и RISC процессоры.
4. Первые ЭВМ, первые устройства ввода и вывода. Учебные машины УМ-3, УМ-2, УМ-1, как примеры организации первых ЭВМ.
5. Машинное представление целых чисел: прямой код, дополнительный код. Выставление флагов знакового и беззнакового переполнения. Флаг знака и флаг нуля. Диапазоны знаковых и беззнаковых чисел. Получение обратного числа через побитовое отрицание и сложение.
6. Машинное представление чисел с плавающей точкой. Числа с плавающей точкой половинной, одинарной, двойной точности. Числа в 80 битном формате FPU блока x86 архитектуры. Специальные числа: бесконечности, неопределённого значения (NaN). Понятие машинного 0, проблемы округления чисел с плавающей точкой. Округление к ближайшему чётному.
7. Общая структура программы на ассемблере в `masm`. Директивы, указание типа процессора и модели памяти. Секции `.data`, `.data?`, `.code`, `.stack`. Резервирование памяти в секции, конструкции: `db`, `dw`, `dd`, `dq`, `dt`, `real4`, `real8`. Организация повторения при резервировании памяти: конструкция `dup`. Резервирование строки.
Описание констант.
8. формат описания команд процессоров в `masm`. префикс, допустимые типы операндов. Косвенная адресация по одному и по двум регистрам. Конструкция `offset`. Указание размера операнда при обращении в память. Метки в коде, в том числе работа с анонимными метками. Точка входа в программу (метка `start`) оформление головного ассемблерного модуля.
9. Понятие компиляции и линковки ассемблерного кода. Модули на ассемблере и объектные модули. Указание внешних и внутренних имён в модулях, директивы `public` и `extern`. оформление не головного ассемблерного модуля.
10. Регистры архитектуры x86-32. Назначение регистров. Регистры общего назначения, специальные регистры. Обращение к частям регистров.
Регистр флагов и его поля. Инструкции очистки и установки флагов: **cld**, **std**, **clc**, **stc**, **cmc**, **cli**, **sti**. Инструкция сохранения арифметических флагов в регистр `ah`: **lahf**, восстановление **sahf**.
Регистры блока FPU, векторные регистры SSE2.
11. Специальные регистры и их роль: `eip` – счётчик адреса, `cs`, `ds`, `es`, `ss`, `fs`, `gs` – сегментные регистры, регистры `gdt`, `ldt`, `idt` – указатели на таблицы, `tr` – регистр задачи.
12. Инструкция пересылки **mov**. Семейство инструкций условной пересылки по состоянию флагов: **cmovcc**. Инструкция обмена **xchg**.
13. Арифметические инструкции: **add**, **sub**, **inc**, **dec**, **adc**, **sbb**, **neg**. Влияние данных инструкций на арифметические флаги.
14. Инструкции умножения **mul**, **imul**, и деления **div**, **idiv** (только одноаргументный вариант).
Инструкция взятия адреса – **lea**. Использование для быстрого умножения на степени двойки (0,1,2,3) со сложением.
15. Инструкции знакового и беззнакового расширения типов: **movsx**, **movzx**.
Инструкции знакового расширения: **cbw**, **cwd**, **cdq**, **cwde**.
16. Побитовые операции: **and**, **or**, **xor**, **not**. Операции сдвигов: **shl**, **shr**, **sal**, **sar**. Циклические сдвиги: **rcl**, **rcr**, **rol**, **ror**. Влияние всех этих инструкций на арифметические флаги.

17. Инструкции для работы с числами с плавающей точкой в наборе инструкций SSE2. Скалярные операции: **addss**, **addsd**, **subss**, **subsd**, **mulss**, **mulsd**, **divss**, **divsd**. Векторные арифметические операции (упакованные данные) с суффиксом *p*.
Сравнение чисел с плавающей точкой: **comiss**, **comisd**, **ucomiss**, **ucomisd**. Выставление флагов в регистре *eflags* операциями сравнения.
Инструкция горизонтального перемешивания частей векторного регистра **shufps**.
18. Семейство инструкций **cvtXXX2YYY** конвертации чисел с плавающей точкой. Использование FPU для конвертации: инструкция **fld** и **fstp**.
19. Инструкции пересылки чисел с плавающей точкой в векторные регистры: **movss**, **movsd**, **movaps**, **movapd**, **movups**, **movupd**.
Инструкции пресылки в FPU и обратно: **fld** и **fstp**.
20. Инструкция сравнения **cmp** и побитового сравнения **test**.
Инструкция перехода: **jmp**. Инструкция условного перехода **jcc**. Переход по значению регистра *ecx* — **jcxz**, **jecxz**.
Инструкция цикла **loop**.
Допустимые аргументы инструкций переходов.
21. Строковые команды: **movsl**, **cmprsl**, **scasl**, **lodsl**, **stosl**. Влияние флага направления DF, сменв направления.
Префиксы повторения: **rep**, **repe**, **repne**.
22. Работа со стеком. Роль регистра *esp*. Инструкции **push** и **pop**. Допустимые аргументы инструкций **push** и **pop**. Инструкции сохранения и восстановления регистров **pushad** и **popad**. Сохранение и восстановление регистра флагов: **pushfd**, **popfd**.
23. Процедуры и функции. Инструкция **call**. Стековый кадр, роль регистра *ebp*. Стандартный пролог и эпилог функции.
24. Процедуры и функции, стандартные соглашения: *stdcall*, *fastcall*, *cdecl*, *pascal*. Доступ к параметрам и передача параметров, организация локальных переменных, сохраняемые функцией/процедурой регистры, возвращение целочисленного результата, возвращение результата как числа с плавающей точкой, очистка стека.
Особенность передачи параметров по ссылке и по значению.
Синтаксис оформления функции в *masm*.
25. Организация структур и массивов. Конструкция *equ*. Описание аналогов *record* языка Pascal конструкция *struct*.
26. Ассемблерные вставки в FreePascal. Организация совместной линковки FreePascal и объектных модулей. Указание внешних имён функций и процедур в FreePascal и стандартных соглашений.
27. Макросредства *masm*. Директива *macro*, параметры макроса: со значениями по умолчанию, требуемые параметры. Указание параметров в угловых скобках. Конструкции разрыва лексемы *ℓ*. Конструкция экранирования символов: *!*. Объявление макропеременных. Локальные метки и макропеременные. Конец макроса директива *endm*, Досрочный выход из макроса директива *exitm*. Директива *.err*. Печать информационного сообщения при компиляции — директива *echo*.
28. Конструкции определения типов и размеров в *masm*: *type*, *size*, *sizeof*, *length*, *lengthof*. Определение атрибутов параметра: директива *opatr*.
29. Директивы условной компиляции: *if*, *ifb*, *ifdiff*, *ifdiffi*, *ifidn*, *ifidni*, *else*, *elseif*, *endif*.
30. Директивы циклов: *for*, *forc*
31. Макросы ввода вывода, написанные преподавателями для проведения курса: *inint*, *outstrln*, *outword*, ...

32. Микроархитектура процессора: теневые регистры, буфер команд, множественность арифметико-логических устройств, конвейерность устройств. КЭШ память, уровни КЭШ. Трансляция команд процессора во внутренние RISC команды.
33. Конвейерность. Стадии конвейера, возможные причины простоя конвейера (образования «пузырей»), причины сброса конвейера. Механизм предсказания переходов и спекулятивного выполнения.
34. Организация КЭШ памяти. Понятие строки/линии КЭШ. Попадание/промах в КЭШ. КЭШ прямого отображения, ассоциативный КЭШ. Сквозная и отложенная запись в КЭШ. Проблема синхронизации КЭШ для нескольких процессорных ядер/процессоров. Префикс **lock** у инструкций. Инструкции принудительной синхронизации: **clwb**, **clflush**, **sfence**, **lfence**, **mfence**. Семейство инструкций записи в память минуя КЭШ **movntxx**. Инструкции подгрузки в КЭШ: **prefetch0**, **prefetch1**, **prefetch2**.
35. Организация прерываний в архитектуре x86-32. Общие принципы организации обработки прерываний от внешних устройств. Внутренние прерывания, Прерывания и исключения. Компоненты подсистемы обработки прерываний. Таблица IDT, Сегмент TSS. Организация выбора обработчика прерывания. Маскирование прерываний, немаскируемые прерывания. Обработка прерывания на своём стеке и со сменой стека. Типичные пролог и эпилог функции обработчика прерывания. Инструкции: **int** и **iret**.