



Содержание

Используемый компилятор.....	9
Глава 1. Основы	11
1.1. Основные понятия	11
1.1.1. Что такое процессор?	11
1.1.2. Небольшая предыстория	14
1.1.3. Процессоры x86-64	16
1.1.4. Регистры процессоров x86-64	18
1.1.5. Память	19
1.1.6. Работа с внешними устройствами	21
1.1.7. Резюме	21
1.2. Основы ассемблера	22
1.2.1. Немного о языке ассемблера	22
1.2.2. Регистр флагов	23
1.2.3. Команда MOV	24
1.2.4. Формат хранения данных в памяти	26
1.2.5. Команды SUB и ADD	27
1.2.6. Логические операции	27
1.2.7. Сдвиги	28
1.2.8. Работа с флагами процессора	30
1.2.9. Работа со стеком	30
1.2.10. Резюме	30
1.3. Метки, данные, переходы	31
1.3.1. Данные	31
1.3.2. Метки	32
1.3.3. Переходы	35
1.3.4. Безымянные метки	38
1.3.5. Работа с битами	39
1.3.6. Резюме	39
1.4. Изучаем ассемблер подробнее	39
1.4.1. Работа с памятью и стеком	40
1.4.2. Работа с числами на ассемблере	41
1.4.3. Умножение и деление	44
1.4.4. Порты ввода-вывода	46
1.4.5. Циклы	46
1.4.6. Обработка блоков данных	47
1.4.7. Макросы	50

1.4.8. Структуры	52
1.4.9. Работа с MSR-регистрами	53
1.4.10. Команда CPUID	54
1.4.11. Команда UD2	55
1.4.12. Включение файлов.....	55
1.4.13. Резюме	55

Глава 2. Защищённый режим..... 56

2.1. Введение в защищённый режим	56
2.1.1. Уровни привилегий.....	56
2.1.2. Сегменты в защищённом режиме	58
2.1.3. Глобальная дескрипторная таблица	61
2.1.4. Практика.....	63
2.1.5. Резюме	70
2.2. Прерывания в защищённом режиме	71
2.2.2. Дескрипторы шлюзов	72
2.2.3. Исключения	74
2.2.4. Коды ошибок	76
2.2.5. Программные прерывания.....	77
2.2.6. Аппаратные прерывания	77
2.2.7. Обработчик прерывания	79
2.2.8. Практика.....	80
2.2.9. Резюме	85
2.3. Механизм трансляции адресов.....	85
2.3.1. Что это такое?.....	85
2.3.2. Обычный режим трансляции адресов	87
2.3.3. Режим расширенной физической трансляции адресов.....	91
2.3.4. Обработчик страничного нарушения.....	94
2.3.5. Флаг WP в регистре CR0	95
2.3.6. Практика.....	96
2.3.7. Резюме	102
2.4. Многозадачность.....	102
2.4.1. Общие сведения.....	102
2.4.2. Сегмент задачи (TSS)	103
2.4.3. Дескриптор TSS.....	105
2.4.4. Локальная дескрипторная таблица	105
2.4.5. Регистр задачи (TR)	106
2.4.6. Управление задачами	106
2.4.7. Шлюз задачи	109
2.4.8. Уровень привилегий ввода-вывода.....	109
2.4.9. Карта разрешения ввода-вывода.....	110
2.4.10. Включение многозадачности	110
2.4.11. Практическая реализация	111
2.4.12. Резюме	118

2.5. Механизмы защиты	119
2.5.1. Поля и флаги, используемые для защиты на уровне сегментов и страниц	119
2.5.2. Проверка лимитов сегментов	120
2.5.3. Проверки типов	120
2.5.4. Уровни привилегий	122
2.5.5. Проверка уровня привилегий при доступе к сегментам данных	123
2.5.6. Проверка уровней привилегий при межсегментной передаче управления	124
2.5.7. Шлюзы вызова	125
2.5.8. Переключение стека	128
2.5.9. Использование инструкций SYSENTER и SYSEXIT	129
2.5.10. Практика	130
2.5.11. Резюме	133
Глава 3. ПРОГРАММИРОВАНИЕ В WIN32	134
3.1. Введение в Win32	134
3.1.1. Основные сведения	135
3.1.2. Память в Win32	135
3.1.3. Исполняемые компоненты Windows	136
3.1.4. Системные библиотеки и подсистемы	137
3.1.5. Модель вызова функций в Win32	138
3.1.6. Выполнение программ в Win32: общая картина	138
3.1.7. Практика	139
3.1.8. Резюме	147
3.2. Программирование в третьем кольце	148
3.2.1. Общий обзор	148
3.2.2. Работа с объектами	149
3.2.3. Работа с файлами	149
3.2.4. Обработка ошибок API-функций	152
3.2.5. Консольные программы	152
3.2.6. GUI-программы	153
3.2.7. Динамически подключаемые библиотеки	156
3.2.8. Обработка исключений в программе	159
3.2.9. Практика	162
3.2.10. Резюме	171
3.3. Программирование в нулевом кольце	171
3.3.1. Службы	172
3.3.2. Общий обзор	173
3.3.3. Driver Development Kit (DDK)	174
3.3.4. Контекст потока и уровни запросов прерываний	175
3.3.5. Пример простого драйвера	176
3.3.6. Строки в ядре Windows	179

3.3.7. Подсистема ввода-вывода.....	180
3.3.8. Практика.....	186
3.3.9. Резюме.....	201

Глава 4. LONG MODE..... 202

4.1. Введение в long mode.....	202
4.1.1. Общий обзор.....	202
4.1.2. Сегментация в long mode.....	204
4.1.3. Механизм трансляции страниц.....	205
4.1.4. Переход в long mode.....	205
4.1.5. Практика.....	206
4.1.6. Резюме.....	208
4.2. Работа с памятью в long mode.....	208
4.2.1. Общий обзор.....	209
4.2.2. Страницы размером 4 Кб.....	209
4.2.3. Страницы размером 2 Мб.....	211
4.2.4. Страницы размером 1 Гб.....	212
4.2.5. Регистр CR3.....	213
4.2.6. Проверки защиты.....	214
4.2.7. Практика.....	214
4.2.8. Резюме.....	221
4.3. Прерывания в long mode.....	221
4.3.1. Дескрипторы шлюзов.....	221
4.3.2. Таблица IDT, 64-битный TSS и механизм IST.....	222
4.3.3. Вызов обработчика прерывания.....	223
4.3.4. Практика.....	224
4.3.5. Резюме.....	230
4.4. Защита и многозадачность.....	230
4.4.1. Сегменты.....	231
4.4.2. Шлюзы вызова.....	231
4.4.3. Инструкции SYSCALL и SYSRET.....	232
4.4.4. Многозадачность.....	233
4.4.5. Практика.....	235
4.4.6. Резюме.....	238

Глава 5. ПРОГРАММИРОВАНИЕ В WIN64..... 239

5.1. Введение в Win64.....	239
5.1.1. Преимущества и недостатки.....	239
5.1.2. Память в Win64.....	240
5.1.3. Модель вызова.....	240
5.1.4. Режим совместимости.....	242
5.1.5. Win64 API и системные библиотеки.....	242
5.1.6. Практика.....	243
5.1.7. Резюме.....	244

5.2. Программирование в Win64	244
5.2.1. Изменения в типах данных	245
5.2.2. Выравнивание стека	245
5.2.3. GUI-приложения	246
5.2.4. Программирование драйверов	250
5.2.5. Отладка приложений в Win64	254
5.2.6. Резюме	254
Глава 6. МНОГОПРОЦЕССОРНЫЕ СИСТЕМЫ	255
6.1. Работа с APIC	255
6.1.1. Общий обзор	255
6.1.2. Включение APIC	256
6.1.3. Local APIC ID	257
6.1.4. Локальная векторная таблица	257
6.1.5. Local APIC Timer	259
6.1.6. Обработка прерываний	261
6.1.7. Работа с I/O APIC	263
6.1.8. Практика	266
6.1.9. Резюме	270
6.2. Межпроцессорное взаимодействие	270
6.2.1. Общий обзор	270
6.2.2. Межпроцессорные прерывания	271
6.2.3. Синхронизация доступа к данным	273
6.2.4. Инициализация многопроцессорной системы	275
6.2.5. Практика	276
6.2.6. Резюме	280
ПРИЛОЖЕНИЯ	281
Приложение А. MSR-регистры	281
А.1. Регистр IA32_EFER	281
А.2. Регистры, используемые командами SYSENTER/SYSEXIT	281
А.3. Регистры, используемые командами SYSCALL/SYSRET	282
А.4. Регистры APIC	282
А.5. Регистры для управления сегментами в long mode	283
А.6. Вспомогательные регистры	283
Приложение Б. Системные регистры	283
Б.1. Регистр CR0	283
Б.2. Регистры CR2 и CR3	285
Б.3. Регистр CR4	286
Б.4. Регистры GDTR и IDTR	287
Б.5. Регистры LDTR и TR	288
Б.6. Регистр флагов	288
Б.7. Регистр CR8	290

Приложение В. Системные команды	290
В.1. Работа с системными регистрами	290
В.2. Системные команды.....	293
В.3. Работа с кэшем процессора	295
В.4. Дополнительные команды	295
Алфавитный указатель	297



Используемый компилятор

При компиляции всех примеров, приведённых в этой книге, необходимо использовать FASM. Почему я выбрал именно этот компилятор? По очень многим причинам.

1. Максимальный набор поддерживаемых команд. FASM поддерживает весь (или почти весь) набор команд x86-64.
2. Наличие Linux- и Windows-версий, а также поддержка широкого списка выходных файлов. Можно компилировать программы для Windows (формат PE) и для Linux (формат ELF).
3. Гибкий синтаксис и отсутствие совершенно бесполезных, но обязательных директив (например, .386, .486 и т. п.).
4. Полный контроль над размещением данных в исполняемом файле.
5. Мощнейший макросный движок, с помощью которого можно создать практически любой макрос и изменить текст программы и сам язык до неузнаваемости.
6. При компиляции через командную строку нет необходимости указывать множество опций компиляции. Для того чтобы получить исполняемый файл нужного типа, достаточно задать все необходимые параметры в самом исходнике.
7. Windows-версия FASM поставляется вместе с IDE, благодаря которой можно скомпилировать программу нажатием одной кнопки (или пункта меню).

Компилятор FASM можно найти на компакт-диске, прилагающемся к книге (папка fasm). Последнюю версию этого компилятора также можно бесплатно скачать на сайте flatassembler.net.