3. Исполняемый файл. Этапы запуска исполняемого файла. Функция main. (ОК)
План ответа:
 представление о формате исполняемого файла;
 загрузка файла в память;
 загрузка библиотек;
 настройка ссылок;
 планирование процесса;
 абстрактная память процесса;
 заголовки функции main согласно стандарту C99;
 значение, возвращаемое функцией main;
 аргументы функции main.

1. Функция main – точка входа программы на Си, обязательная составляющая каждой завершённой программы.
1.1. Заголовки функции main согласно стандарту C99:
	int main(void). Приложение, скомпилированное с таким заголовком, не берёт параметры во время запуска.
	Применяется в основном в случаях, когда параметры командной строки не нужны при работе с программой.
	int main(int argc, char** argv). Этот заголовок позволяет применять параметры командной строки при работе программы.
1.2. Возвращается код ошибки выполнения программы согласно соглашениям операционной системы.
	Они, как правило, описаны в константах, содержащихся в заголовочном файле <errno.h>.
	Если возвращается значение <=0, значит, системных ошибок не было.
	Считается хорошим тоном несистемные ошибки выполнения программы кодировать отрицательными значениями,
	нормальное выполнение программы – нулём.
1.3. Параметр argc функции main содержит число введённых параметров командной строки,
	argv – это массив строк, содержащих эти значения. Строки с аргументами в командной строке разделяются пробелами и знаками табуляции.
	argv[0] всегда является именем приложения, поэтому всегда argc>=1. Согласно стандарту, argv[argc]==NULL.
Стандарт C99 требует от компилятора возможность обработать по крайней мере 32767 аргументов командной строки, что навряд ли под силу какой-либо ОС.

2.1. Его структура состоит из следующих частей:
	Заголовочные блоки (не менее одного). Как правило, там содержится информация о том, как правильно отображать функции в память.
	Секция .text. В ней содержится исполняемый код.
	Секция .bss. В ней хранятся неинициализированные глобальные и статические переменные.
	Секция .data. В ней содержатся инициализированные глобальные и статические переменные для чтения и изменения.
	Секция .rodata. В ней содержатся данные только для чтения.
	Таблица импорта. Через неё осуществляется связь с внешними библиотеками и вызов функций или переменных по адресам.
	???Таблица экспорта. Через неё указывается, какие адреса из модуля могут быть импортированы внешней программой или внешним модулем.
	???Таблица перемещений. Через неё вычисляются адреса всех данных и функций, используемых в программе.
	???Метаданные. Включают информацию, относящуюся к сборке продукта, включая её манифест.
2.2. Превращение исполняемого файла в процесс. Этапы:
	Считывание программы из диска в ОЗУ.
	Загрузка динамических библиотек.
	Настройка ссылок.
	Планирование процесса.
	При запуске процессу выделяется "абстрактная память" и помещает в неё исполняемый файл.
	Процесс считает, что в этой памяти содержатся только его данные

В абстактной памяти содержимое программы представлено следующим образом
	Нулевой адресс
	Исполняемы файл
		код
		данные

Таблица импорта
Библиотеки
Код
Данные
Куча
Стек