Контрольные вопросы:

1. Регистрация пользователя в системе происходит с использованием команды useradd. /etc/passwd - хранит информацию о пользователях, доступен для просмотра всем пользователям.

/etc/shadow - хранит информацию о пользователях и их паролях, доступен только суперпользователю.

1. Процесс загрузки операционной системы можно разбить на 5 этапов, причём первые три зависят от материнской платы.

Первый этап: включение компьютера  
UEFI + GPT

Интерфейс расширяемой прошивки EFI (позднее унифицированный UEFI) – это разработка Intel. Он был создан потому, что BIOS имеет ряд недостатков. К ним можно причислить работу в 16-битном режиме, последовательную проверку устройств и загрузку только через MBR.

UEFI работает в 32- и 64-битном режиме, поддерживает таблицу разделов GPT, может иметь красивый графический интерфейс, оболочку EFI Shell и управление с помощью мыши. К тому же опрос оборудования происходит гораздо быстрее.

После проверки устройств, UEFI может самостоятельно обратиться к таблице разделов GPT. В ней её интересует EFI раздел. Обычно под него выделяется 520-540 Мбайт. Такой размер связан с тем, что ранние версии EFI могли выдавать ошибку при работе с меньшим объёмом. В разделе EFI обязательно должна быть папка EFI/BOOT, в ней находится загрузчик по умолчанию.

Второй этап: загрузчик

Загрузчик подготавливает систему к загрузке ядра операционной системы.  
GRUB  
Он позволяет не только запустить ядро Linux или другой операционной системы, но ещё и сам по себе способен выполнять некоторые операции. Он имеет собственную командную строку и в теории позволяет устранить собственные ошибки.  
EFI BOOT STUB

Начиная с версии 3.3 ядра Linux можно упаковать ядро в .efi файл. Иными словами, загрузчиком будет выступать само ядро Linux.

Третий этап: ядро

Дальше выполняется загрузка ядра Linux. Ядро лежит в папке boot под названием vmlinuz. «vm» в начале говорит о том, что ядро будет размещено в виртуальной памяти. А «z» на конце означает то, что ядро запаковано в архив и требует распаковку.

В процессе запуска ядра настраивается оборудование, запускается менеджер памяти, планировщик задач и многое другое, необходимое для функционирования операционной системы. Ядро Linux монолитное, драйверы устройств хранятся прямо в нём. Однако, большая часть периферийных устройств не имеет драйверов в ядре Linux. Вместо этого в оперативной памяти распаковывается ещё один файл во временной файловой системе – initrd.img.

Четвертый этап: запуск пользовательских служб

Ядро загрузилось и работает. Однако, для того чтобы можно было что-то сделать в Linux, управление получает подсистема инициализации и управления службами. Обычно этот процесс имеет идентификационный номер PID-1.   
Systemd – это самая популярная система инициализации. Запускаются десятки служб. Большая часть файлов подсистемы представлена конфигурационными файлами, ссылающимися на конкретные утилиты и сервисы, но есть и собственные службы. В комплект утилит Systemd входит супервизор. Он следит за тем, чтобы все критически важные службы работали. Если он заметит, что какой-то службы не хватает, то сразу же запустит её снова. Его также применяют для поддержания работы серверов.

Пятый этап: экранный менеджер и окружение рабочего стола

Дисплейный менеджер отвечает за вход в систему. Хотя и до его запуска проводились некоторые работы, был запущен сервер X11 для обращения к видеокарте и обработки команд управления с мыши и клавиатуры.

После ввода пароля запускается оконный менеджер с библиотеками отображения (обычно GTK3 или Qt) и непосредственно само окружение рабочего стола. Эти службы отвечают за то, чтобы вы могли перемещать окна, перетаскивать файлы, иметь несколько рабочих столов и тому подобное.

1. / - корень. Это главный каталог в системе Linux. Адреса всех файлов начинаются с корня, а дополнительные разделы, флешки или оптические диски подключаются в папки корневого каталога. Только пользователь root имеет право читать и изменять файлы в этом каталоге.

/bin - содержит основные исполняемые файл, доступные всем пользователям, а также содержит символьные ссылки на исполняемые файлы.

/boot - содержит конфигурационные файлы загрузчика GRUB, образы ядра, файлы Initrd. Файл Initrd - образ этой временной корневой системы, который хранится на загрузочном устройстве.

/dev - содержит файлы устройств. В Linux устройством называется оборудование, которое подключается к системе в процессе загрузки ядра или в процессе работы системы. Эти устройства представляют методы для ввода или вывода информации. Большинство устройств в Linux представляют из себя файлы в особой файловой системе (исключение составляют сетевые карты). И вот эти файлы хранятся в каталоге /dev, куда к ним обращается система для выполнения задач, связанных с вводом/выводом.

/etc - содержит конфигурационные файлы операционной системы и всех сетевых служб.

/home - здесь содержатся домашние каталоги всех пользователей, которые зарегистрированы в системе. В домашних каталогах пользователей хранятся пользовательские файлы, а также пользовательские настройки различных программ.

/lib - здесь находятся различные библиотеки и модули ядра.

/lost+found - этот каталог нужен для хранения испорченных файлов при проблемах с файловой системой, которые были восстановлены после.

/mnt и /media - обычно в этих каталогах содержатся точки монтирования.

/opt - здесь обычно размещаются установленные программы, имеющие большой дисковый объем, или вспомогательные пакеты.

/root - каталог пользователя root.

/sbin - набор утилит для системного администрирования, содержит исполняемые файлы, необходимые для загрузки системы и ее восстановления в различных щекотливых ситуациях. Запускать эти утилиты имеет право только root.

/tmp - каталог, в котором хранятся временные файлы.

/usr - содержит пользовательские программы, документацию, исходные коды программ и ядра.

/var - содержит файлы, которые подвергаются наиболее частому изменению.

1. П
2. Gcc - препроцессирует и компилирует C и C++ исходные файлы, затем ассемблирует и компонует их вместе.

Компилирует несколько исходных файлов в исполняемый:

gcc {{source1.c}} {{source2.c}} -o {{executable}}

Включает предупреждения и отладочные символы в выходном файле:

gcc {{source.c}} -Wall -Og -o {{executable}}

Подключает библиотеки, находящиеся по другому пути:

gcc {{source.c}} -o {{executable}} -I{{header\_path}} -L{{library\_path}} -l{{library\_name}}

Компилирует исходный код в ассемблерные инструкции:

gcc -S {{source.c}}

Компилирует исходный код в объектный файл (без компоновки):

gcc -c {{source.c}}