

Тема. Создание инструментальной платформы на базе GNU/Linux

Составитель: Фёдоров С. А.

Редакция от: 09.02.18

Получаемая квалификация

- Создание инструментальной платформы на базе GNU/Linux.
- Работа с менеджером пакетов.
- Установка *свежих* (тестируемых и нестабильных) версий инструментов разработки ПО на *стабильную* систему.

Общее задание

При выполнении лабораторной работы необходимо следовать руководству «Создание инструментальной платформы на базе GNU/Linux».

1. Провести установку дистрибутива Debian GNU/Linux 9.x (stretch) как отдельно стоящей ОС (глава 1).
Можно использовать другой дистрибутив *кроме*:
 1. дистрибутивов, предназначенных не для разработки ПО, а для пользователей бытовыми системами – семейство дистрибутивов Ubuntu, Mint и т. д.;
 2. дистрибутивов, предназначенных для тестирования нового ПО и подходов в организации дистрибутивов: Fedora, openSUSE,Дистрибутив должен позволять приобрести следующую квалификацию:
 3. Эффективная настройка системы через командную строку (WiFi, установка пакетов и т. п.).
 4. Эффективная работа со сложноструктурированными текстами *с клавиатуры* – Vim.
 5. Эффективная работа с клиентами приложений в графическом режиме *с клавиатуры* (менеджеры окон awesome, i3 и т. п.).
 6. Работа в системе одновременно с пакетами как из стабильных веток, так и из тестируемых и нестабильных.
 7. Использование *стабильной версии* дистрибутива вместе с *последними* инструментальными средствами (например, GCC, Vim и т. д.).
2. Сделать загрузчик GRUB запускаемым по умолчанию в системе (глава 2).
3. Освоить менеджер пакетов aptitude (см. меню, помощь («?») и руководство по aptitude).
4. Провести настройку системы как инструментальной платформы (глава 3), установив *свежие* версии инструментов разработки ПО (*из тестируемой или нестабильной ветки*) на *стабильную* версию дистрибутива.

Отчётность

Подготовить в электронном виде документ (не отчёт) со следующей информацией.

1. Процессор инструментальной платформы:
 1. *модель* процессора;
 2. архитектура;

3. микроархитектура;
 4. кодовое имя микроархитектуры;
 5. число ядер;
 6. число потоков, обрабатываемых на кристалле;
 7. размер кэш-памяти всех уровней;
 8. связность кэш-памяти последнего уровня (общая/не общая для какого числа ядер).
2. Название установленного дистрибутива, его версия.
 3. Используемый менеджер окон (должен использоваться плиточный менеджер окон, позволяющий *эффективно работать с клавиатуры*, например, awesome, i3 и т. д.);
 4. Сложности, возникшие при создании инструментальной платформы, и способы их преодоления.