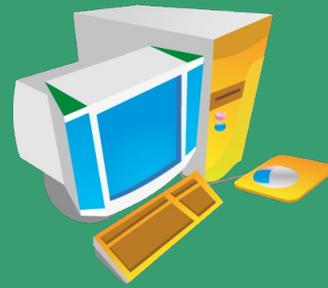


Гуляев Г.М.

Роль и место СПО в современном мире

АлтГТУ, кафедра САПР, altailand.ru



Об авторе

- ❖ Кандидат физ.-мат наук, доцент кафедры САПР АлтГТУ
- ❖ Разработчик ПО: в прошлом автор ряда программ для Windows, в том числе редактора текстов **TextViewer** и системы тестирования людей **AnyTest**
- ❖ Автор проекта с открытым исходным кодом **aomai.ru** - системы дистанционного обучения студентов
- ❖ Автор ПО и владелец интернет-проектов: **altailand.ru**, **altai-flowers.ru** (Цветы Алтая), **foto.altailand.ru** (Фотографии Алтая), **trip.altailand.ru** (Путешествия по Алтаю), **soft.altailand.ru** (Эффективный софт)
- ❖ Контакты:
 - email: **info@altailand.ru**, телефон **8-913-269-8630**
 - ICQ: **9851512**, skype: **ggm2333**
 - форум: **altailand.ru/forum**

Что такое СПО?

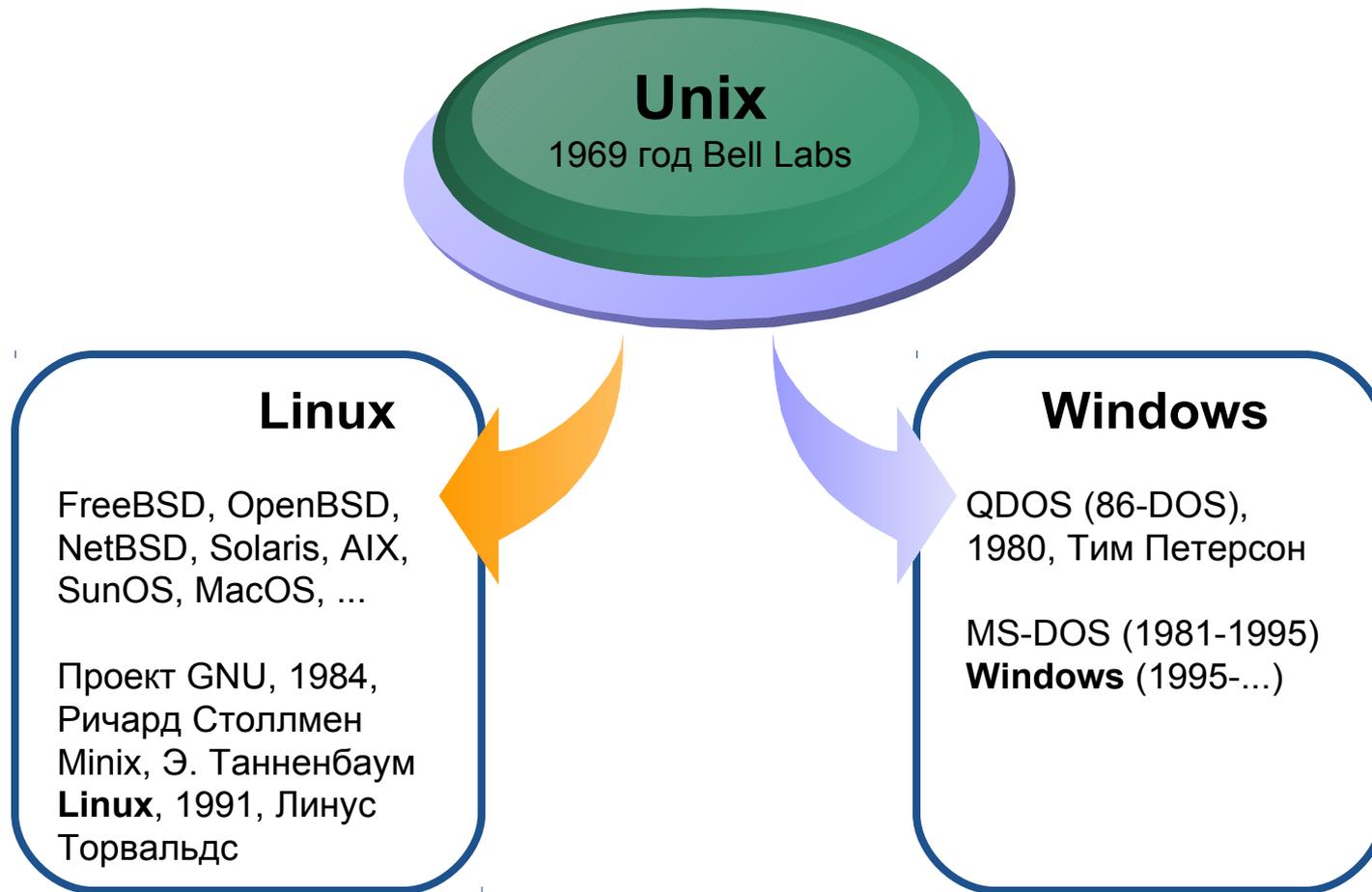
- ❖ **Ключевое слово: свободное. Свобода изучения и исследования, свобода использования и копирования**
- ❖ **Свободное - не обязательно бесплатное и наоборот**
- ❖ **Свобода - это больше возможностей и меньше ограничений, но свобода - это еще ответственность выбора и риск ошибок**
- ❖ **Главным условием для развития (прогресса) является свободное получение и распространение информации**
- ❖ **Искусственные ограничения этого принципа ради выгоды или по иным причинам не могут быть эффективными, ибо они направлены против вектора прогресса**

Кто создает и использует СПО?

- ❖ Люди, мировое сообщество, мы сами
- ❖ Мы разделены расстояниями, странами, языками, но у нас есть общая история, культура, идеология и философия
- ❖ Хотите присоединиться к нам? Мы рады, но будьте добры познакомиться с нашими ценностями:

Эрик Реймонд «Искусство программирования для Unix»,
Вильямс, 2005. - 544 с

- ❖ Принципы:
 - свободно исследуй, изменяй, копируй и распространяй
 - используй готовое или напиши свое
- ❖ Идеология СПО стимулирует каждого принять участие в его создании и тем самым способствует прогрессу в области информационных технологий



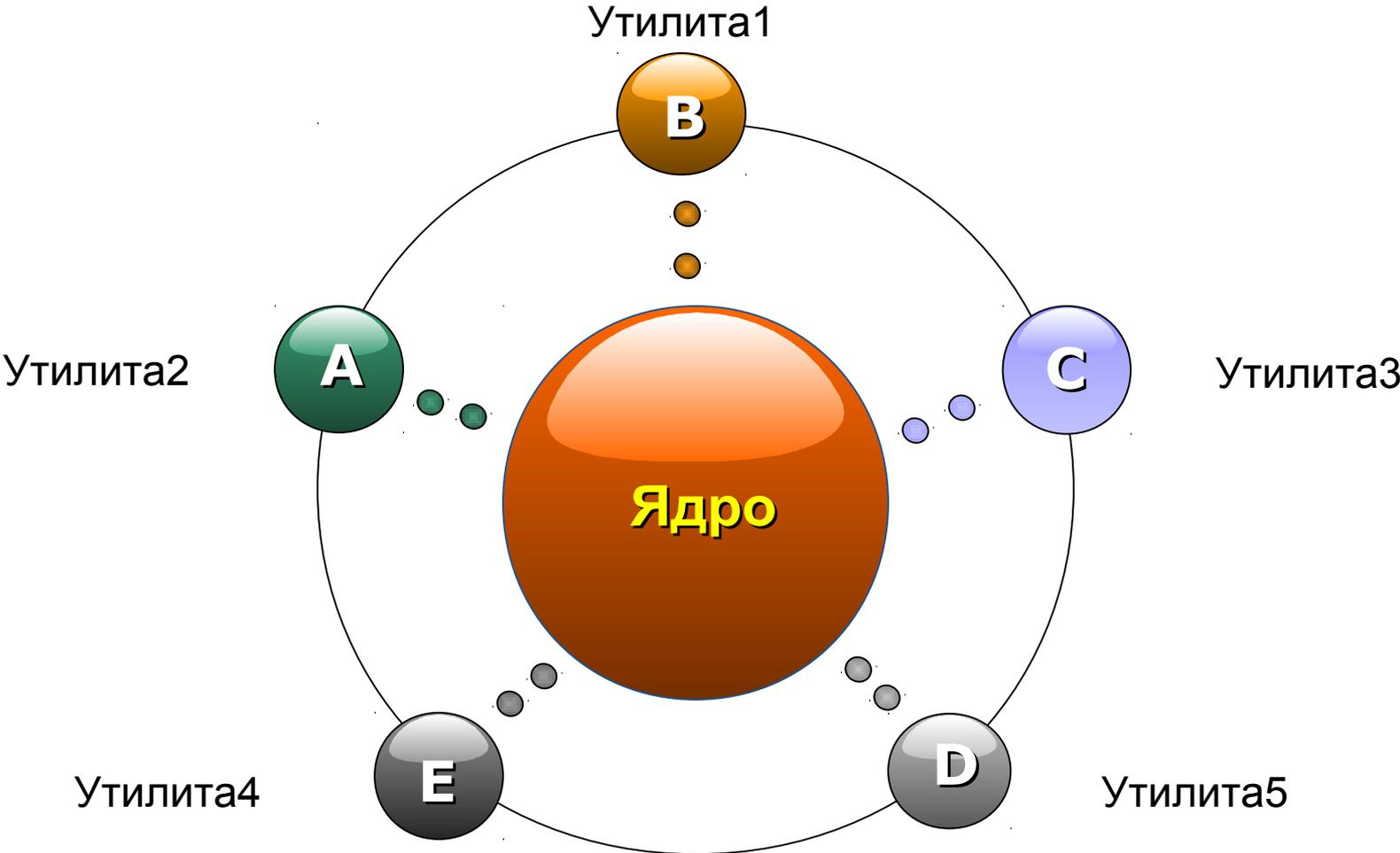
Распространение ПО (терминология)



Лицензии свободного ПО

	GPL	BSD	MIT	Mozilla	Apache
Имя автора	✓	✓	✓	✓	✓
Изм. файлы отмечаются	✓	--	--	--	--
Изм. Файлы — новое имя	--	--	--	✓	✓
На условиях исходного	✓	--	--	✓	--
Территория указывается	--	--	--	✓	--
Отсутствие гарантий	✓	✓	✓	✓	✓

Структура Unix систем



Что входит в Linux?

- ❖ Ядро (более 15 млн строк кода на языке C) - группа разработчиков, координатор Линус Торвалдс
- ❖ Множество маленьких программ, разработанных разными авторами в рамках проектов GNU, BSD и других
- ❖ Необязательная графическая оболочка X Window (**Unity, Gnome, KDE, XFCE, LXDE**)
- ❖ Разнообразное пользовательское прикладное ПО для графической оболочки
- ❖ Имея доступ к исходному коду, практически любое ПО, написанное для **unix**, можно адаптировать для **Linux**

Какой Linux выбрать?

- ❖ Все **unix**-системы построены на одних и тех же принципах, изучая любую из них мы изучаем их все
- ❖ С точки зрения внутреннего содержания абсолютно нет никакой разницы между различными **linux**-дистрибутивами: одно ядро и те же программы
- ❖ Разница между **FreeBSD** и **Debian** в ядре, между **Debian** и **RedHad** в пакетах, между **Debian** и **Ubuntu** - никакой разницы нет (это разные сборки одной и той же системы)
- ❖ При выборе системы важную роль играет размер ее сообщества, в этом смысле **Ubuntu** имеет неоспоримое преимущество:
ubuntu.ru, ubuntu.com и т.д.

Причины для отказа от Windows

- ❖ Она небезопасна. Безопасность не учитывалась при ее проектировании. Как следствие, вирусы и дорогое антивирусное ПО
- ❖ Финансовые затраты. Базовая система не содержит необходимого ПО, либо содержит ПО с ограниченным функционалом. Цель - заставить покупать полнофункциональное ПО
- ❖ Закрытый код скрывает ошибки. Ошибки в ПО тем менее вероятны и тем быстрее исправляются, чем больше людей могут видеть исходный код
- ❖ Закрытая система не годится для обучения, поскольку она препятствует исследованию своего внутреннего устройства

Типичные вопросы при переходе с Windows на Linux

- ❖ Я больше не смогу запускать мои приложения?
 - 1) есть **wine**
 - 2) множество свободных программ
 - 3) современное ПО имеет веб-интерфейс
- ❖ В моей работе никак нельзя без **Windows**.
 - 1) двойная загрузка
 - 2) виртуальная машина в **Linux**
- ❖ Иногда мне нужна помощь и поддержка.
 - 1) большое **Linux**-сообщество
 - 2) постоянная поддержка и обновления СПО
 - 3) поддержка от **Microsoft** не более чем миф

Поддержка или как решать проблемы

- ❖ Проблема с оборудованием или ПО, возникшая у вас, скорее всего, уже была обнаружена другими пользователями и имеет свое решение
- ❖ Типичный алгоритм:
 - 1) поиск в интернет
 - 2) выбор правильной рекомендации
 - 3) применение
- ❖ Для успешного поиска проблема должна быть локализована и точно сформулирована
- ❖ Не все рекомендации стоит применять - нужно иметь достаточный уровень понимания для того, чтобы выбрать подходящее для вас решение

Чему следует и не следует учиться?

- ❖ Область ИТ (информационных технологий) быстро развивается и непрерывно изменяется
- ❖ Не следует учиться внешнему виду (интерфейсу) программы - он быстро меняется со временем по желанию разработчика и в угоду моде
- ❖ Следует учиться логике, принципам, правилам и законам, лежащим в основе функционирования аппаратного и программного обеспечения
- ❖ Не следует учиться закрытым стандартам и программам, для которых можно найти открытые аналоги
- ❖ Следует учиться технологиям и программам, которые поощряют исследование и изменение самих себя

Спасибо за внимание!

www.altailand.ru

