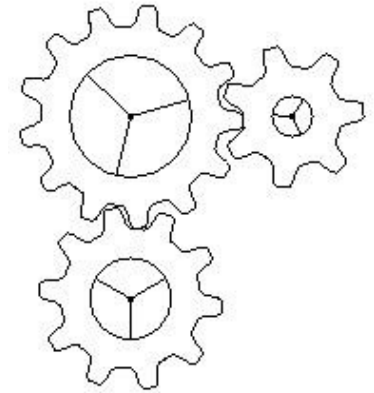
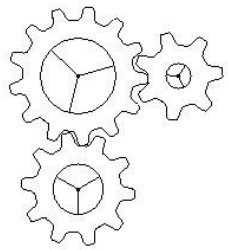


Исследования в области операционных систем Ждать ли рассвета?



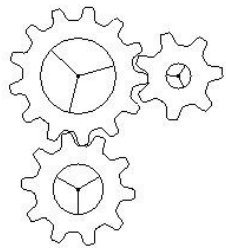
Петр Новодворский, ВМиК
МГУ

<nidd@cs.msu.ru>



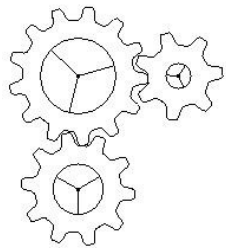
Цели доклада

- Анализ истории развития ОС
- Попытка заглянуть вперед
- Рассказать, что такое исследование ОС и заинтересовать аудиторию
- Библиография в конце тезисов, возможно, заслуживает большего внимания чем тезисы и доклад



Три касты

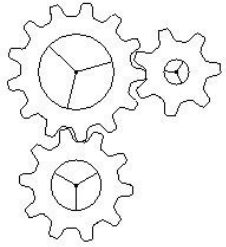
- **Разработчики ОС (инженеры)**
 - Linus Torvalds, Ingo Moolinar, Andrew Morton and Linux developers, Theo de Raadt and OpenBSD, ??? the Windows Vista Developers
- **Разработчики компьютерного оборудования**
 - David Patterson, Roger Wilson, Steve Furber, John Hennessy
- **Исследователи ОС (типа ученые)**
 - Andrew Tannenbaum, Jay Lapreau, Steve Gribble, Ken Thompson, Jochen Liedke, Rob Pike, другие



Разработчики ОС

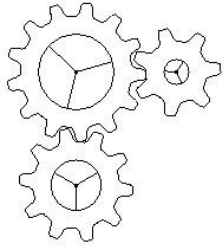
- Linus Torvalds, Ingo Moolinar, Andrew Morton and Linux developers, Theo de Raadt and OpenBSD, ??? the Windows Vista Developers
- Место обитания: разработческие подразделения в компаниях, подрядчики, студенты в свободное время
- Результаты работы: операционные системы которые мы используем

Разработчики компьютерного оборудования



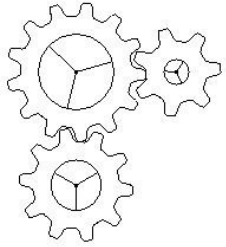
- David Patterson, Roger Wilson, Steve Furber, John Hennessy
- Место обитания: IBM, Intel, ARM, MIPS.
- Результаты работы: архитектуры PowerPC, IA-32, IA-64, AMD64^WEMT64T, ARM воплощенные в кремнии

Исследователи ОС (типа ученые)

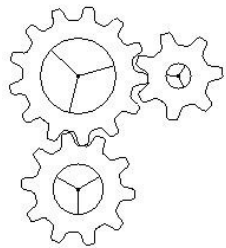


- Andrew Tannenbaum, Jay Lapreau, Steve Gribble, Ken Thompson, Jochen Liedke, Rob Pike, другие
- Место обитания: университеты, государственные лаборатории, лаборатории в компаниях (HP Labs, Intel research, IBM TJ Watson lab).
- Результаты работы: публикации, примеры реализации идей.
- Чем занимаются эти люди??

Чем занимаются исследователи ОС?



- Решают проблемы: (If you found a problem, go and fix it!)
 - Вырабатывают методы для выжимания максимума из имеющегося оборудования
 - Помогают разработчикам оборудования создавать новые процессоры и шины
 - Создают новые, более гибкие в разработке и использовании архитектуры ОС
 - Вырабатывают методы исправления ошибок в существующих ОС

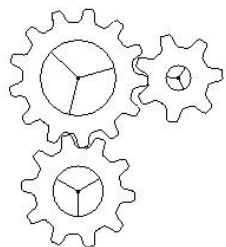


Анализ истории ОС

- Три ветви развития:
 - Новые идеи в исследованиях в области ОС
 - Новшества в популярных коммерческих ОС (commodity operating systems)
 - Развитие архитектуры компьютерного оборудования
- Исследуя эти ветви попытаемся понять как взаимодействуют три группы: разработчики ОС, исследователи ОС и разработчики компьютерного оборудования.
- Тезис: когда эти три группы работают вместе, получаются наиболее эффективные решения

Ранние мейнфреймы от IBM

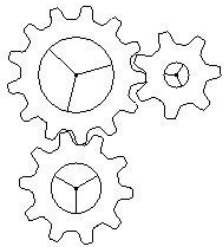
Виртуальные машины до н. э.



- В основном все делается разработчиками оборудования и их пользователями (учеными, но не исследователями ОС)
- Чтобы выполнить одну задачу нам нужна одна машина
- Чтобы выполнить N задач нам нужно N машин
- Как выполнить N задач на одной машине?

Ранние мейнфреймы от IBM

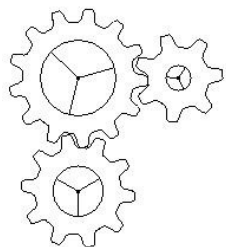
Виртуальные машины до н. э.



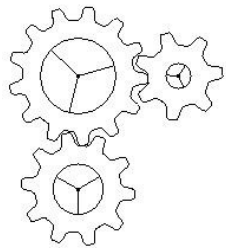
- В основном все делается разработчиками оборудования и их пользователями (учеными, но не исследователями ОС)
- Чтобы выполнить одну задачу нам нужна одна машина
- Чтобы выполнить N задач нам нужно N машин
- Как выполнить N задач на одной машине?
- Многозадачность?
-

Ранние мейнфреймы от IBM

Виртуальные машины до н. э.

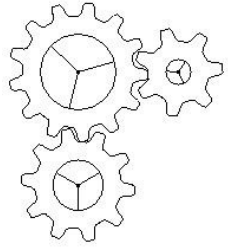


- В основном все делается разработчиками оборудования и их пользователями (учеными, но не исследователями ОС)
- Чтобы выполнить одну задачу нам нужна одна машина
- Чтобы выполнить N задач нам нужно N машин
- Как выполнить N задач на одной машине?
- ■ Многозадачность?
- Виртуализуем машину



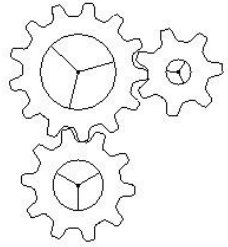
Multics

- Группа разработчиков из General Electric разрабатывал машину GE 645
- Единственная распространенная операционная система для современных машин была у IBM
- GE не хотел платить IBM
- Проект Multics
 - Оборудование от GE
 - Код от Bell Labs
 - Идеи от MIT
- В проекте Multics были заложены те основные механизмы управления ресурсами в ОС, которые используются нами и по ныне:
 - Понятие процесса
 - Защита памяти
 - Понятие файла
 - Динамическое линкование



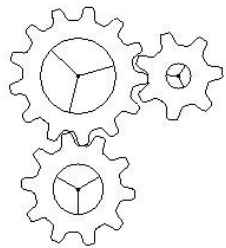
UNIX

- Unix (или Unics, Uniplexed Information and Computing Service) is a castrated Multics (Multiplexed Information and Computing Service)
- Разработчики Unix выжали по-больше из данного им оборудования и из идей Multics, добавив полноценную файловую систему и взаимодействие процессов
- Внимательно приглядевшись к UNIX, можно понять, что это тот же самый Multics, хотя и более легковесный.
- Во время создания UNIX разработчики не координировались с DEC (Unics разрабатывался на DEC PDP-7)
- С другой стороны в Bell Labs были люди наделенные инженерными и исследовательскими талантами одновременно (в той или иной степени), поэтому разработку UNIX можно назвать взаимодействием ученых и инженеров
- (ну и вообще, в те светлые времена все ученые были инженерами и наоборот)



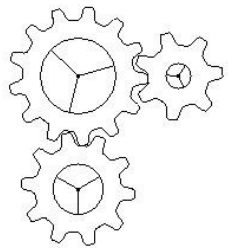
VMS/OpenVMS

- В этом зале сидят люди, которые смогут сказать больше чем я
- VAX и VMS разрабатывались вместе, особенности архитектуры VAX позволяли VMS реализовать наследования capabilities
- Непревзойденная изоляция
- Расширяемость
- Из-за неумелого менеджмента DEC, Digital, Compaq и HP, VMS медленно умирает



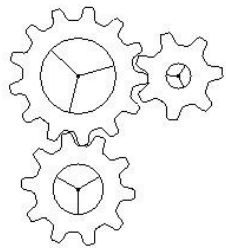
Новый Unix

- В 80-ых и 90-ых добавилось несколько новшеств значительно изменивших методы программирования и администрирования UNIX, но незначительно изменивших его архитектуру. Возможно, именно из-за этого эти методы выглядят так неуклюже под UNIX
- Новшества в программировании
 - Нити (легковесные процессы)
 - Межпроцессное взаимодействие
- Появились новые возможности изоляции процессов
 - Pivot_root (80-ые)
 - Jail (90-ые)
- С таким новшеством как стек протоколов TCP/IP обошлись очень аккуратно и его удалось вписать в концепцию UNIX довольно хорошо



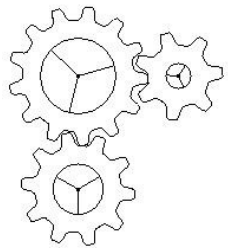
Windows NT

- Новый UNIX с другим API, архитектура та же
- Говорят Microkernel, на самом деле тот же Monolithic
- Все то же самое
- Разрабатывалась для оборудования, а не вместе с оборудованием



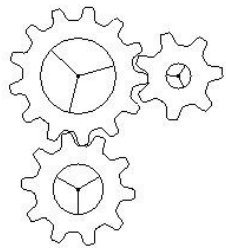
Пропущенные ОС

- ОС про которые большинство почти ничего не знает
 - Mach
 - Plan9
 - SPIN
 - Pebble
 - Denali
 - FluxOS
 - ...
- Исследовательские ОС: создаются для того, чтобы проверить на практике новые идеи.



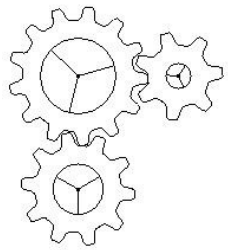
Зачем новые идеи

- Все и так работает и все довольны
- “Новый Unix” – старый UNIX с заплатками, неуклюжими и неудобными, работать с ними невозможно
- Изменения в оборудовании требуют изменения архитектуры системы. UNIX изначально был написан для однопроцессорных машин (в отличие от Multics, не забываем про букву U в начале названия).
- Поддержка SMP в ядре монолитной ОС затрагивает все подсистемы включая файловые системы и системы управления памяти (плохо)
- Появляются такие новшества в оборудовании как multi-core: несколько процессоров делят кэш. Как запускать Cache-hungry приложение с обычным? Как делать эффективное планирование?
- Постоянное перемещение компьютера, переключение от одной сети к другой, переключение устройств ввода вывода.
- Новые идеи постоянно нужны в нашем меняющемся мире.



Mach

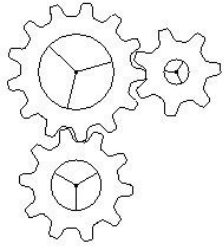
- Цели
 - Расширяемость
 - Поддержка существующих программ
- Описанные цели вообще говоря противоречат друг другу
- Новые идеи позволили достигнуть целей:
 - Разделяется понятие процесса и адресного пространства. Адресное пространство – task, процесс – thread.
 - Взаимодействие между процессами
- ОС Mach: на микроядре работает UNIX сервер с UNIX приложениями, рядом с ним приложения которые свободны от наследия UNIX
- Разработка велась без сотрудничества с разработчиками процессоров (хотя компании разрабатывающие процессоры вкладывали в проект деньги)
- В результате: много новых идей, производительность реализации очень низкая
- Разработчики ОС взяли из Mach threads и IPC и **налепили** их на свои UNIXы.



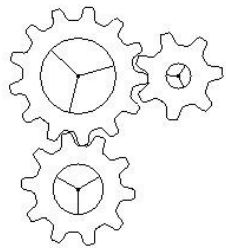
Plan 9

- Попытка переделать UNIX, развить его идеологию.
- Из-за отсутствия поддержки старых интерфейсов ничего не выходит
- Масса идей пропадает

Маленькие исследовательские ОС, новые идеи

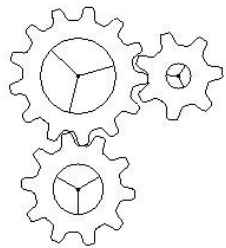


- SPIN – реализация ОС и компонентов на type-safe языке, тем самым достигается изоляция на семантическом уровне
- Scheduler Activations – более подробные интерфейсы между пользовательскими процессами и планировщиком в ядре, позволяют увеличить скорость работы многопоточных программ в разы
- Nooks – обертки вокруг драйверов внутри ядра отслеживающие потоки информации входящей и выходящей из драйвера. Позволяет отслеживать сбои драйверов и их перезагрузку без ущерба окружающей системе
- Exokernel, L4 – маленькие микроядра дающие ещё больше свободы создателю новых ОС на их базе с возможностью запуска в параллель существующих ОС.
- Pebble – несколько реализаций одного сервиса с одинаковыми интерфейсами работают одновременно. Ядро или родитель процесса решает какую реализацию использует потомок.
- **Никого это не волнует.** Для внедрения этих идей в UNIX требуется изменение архитектуры, вендоры не могут на это пойти.



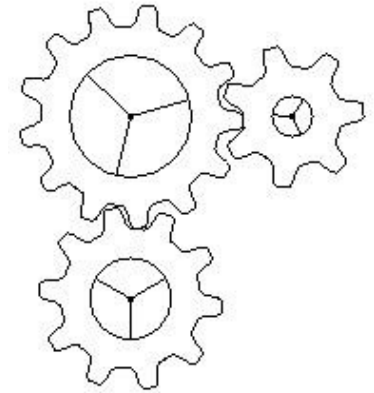
Виртуализация, наша эра

- Совместная разработка компаний разработчиков ПО (Microsoft, VMWare), разработчиков оборудования (VT от Intel) и исследователей (Disco из Univ. of Utah, Denali из Univ. of Washington и Xen от Cambridge)
- Паравиртуализация (Linux на L4)
- Понятно, что виртуализация до н. э. и н. э. Отличаются:
 - Применением
 - Возможностями
- Достигнуты цели Mach: можно запускать существующие ОС и рядом с ними запускать программы с совершенно другой архитектурой



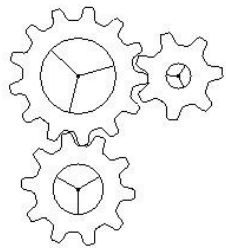
Последствия

- Все наработанные идеи 80-ых и 90-ых не нашедшие подлинного применения тогда смогут найти его сейчас
- Самое время придумать что-нибудь новенькое посмотрев на то, что было придумано тогда
- Актуальные технологии (multicore, например) можно поддержать не латанием дыр а реализацией эффективного доступа к ним



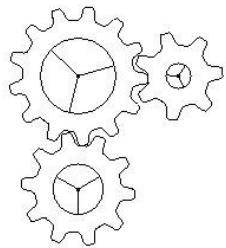
**Вы можете поучаствовать в
улучшении архитектуры ОС!**

А заодно и прославиться!



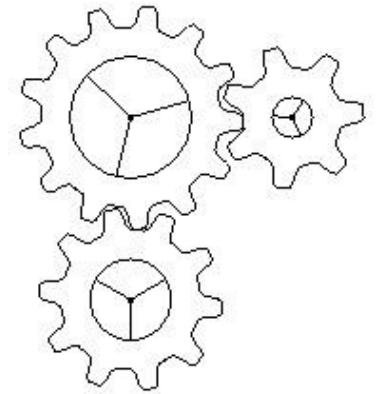
Читайте правильные книги

- FreeBSD. Архитектура и реализация
Маршалл, Кирк, МакКузик, Джордж, Невилл-Нил
 - Перевод плохой, если есть возможность, лучше читать по-английски
- Операционные системы. Разработка и реализация, Таненбаум, Вудхалл
 - Это не современные операционные системы того же автора
- Computer Organization and Design: The Hardware/Software Interface. John Hennessy, David Patterson
 - Этой книжки по-русски нет, но и аналога ей на русском (да и на английском) не существует



Читайте статьи!

- Изучите главные статьи конца прошлого века, вы их найдете в ACM Digital Library (членство в ACM для студентов из России стоит 12\$ в год, professional 43\$)
 - На странице http://myxomop.com/~nidd/bershad_oscourse.html вы найдете подборку статей для курса Брайана Бершада по Операционным Системам
- Основные конференции исследователей ОС: SOSP, USENIX, NSDI, OSDI
- Новая европейская конференция EuroSys
- Следите за статьями публикуемыми на этих конференциях



ПИШИТЕ СТАТЬИ!
Спасибо!