

Задания для курсовых работ по специальности «Вычислительные машины, комплексы, системы и сети» на 2010/2011 учебный год

- 1 Реализация быстрых механизмов синхронизации в пространстве пользователя на примере системного вызова `futex()` в ядре Linux. Работа направлена на исследование механизмов синхронизации между контекстами исполнения с общей памятью, например потоками, не требующих, для проверки занятости ресурса в критической секции, переключения контекста, в связи обращением к ядру через механизмы системных вызовов;
- 2 Исследование механизмов аутентификации и авторизации в GNU/Linux через сетевую службу Active Directory под управлением Windows Server. Работа направлена на исследование механизмов локальной проверки пароля при входе в систему и получению авторизационной информации о пользователях и группах через сетевую службу Active Directory;
- 3 Влияние алгоритмов планирования ввода-вывода на производительность вычислительной системы. Целью работы является исследование влияния различных планировщиков ввода-вывода в Linux на производительность вычислительной системы, поиск компромиссных вариантов настройки операционной системы в зависимости, решаемых задач;
- 4 Реализация работы с образом виртуального жёсткого диска VirtualBox в виде файловой системы в пространстве пользователя. Исследование направлено на изучение формата образа файловой системы и механизмов реализации файловой системы в виде службы. Использование такого механизма предусмотрено модулем FUSE под Linux, и применяется, в частности, для создания зашифрованных файловых систем.
- 5 Особенности выбор компилятора для сборки ядра операционной системы. Работа направлена на выявление текущих тенденций в области использования компиляторов, пригодных или не пригодных по тем или иным причинам для сборки ядер операционных систем. Предполагается исследовать особенности тех версий семейства компиляторов GCC, которые используются для сборки ядер ОС в текущих выпусках дистрибутивов GNU/Linux, исследовательской ОС Minix 3 и других Unix-подобных систем и оценить причины и критерии, на основании которых выбраны именно эти версии компилятора.
- 6 Использование системного кеша ключей для доступа к сетевым службам. В работе предполагается исследование возможностей пакета `keyutils` и кеша удостоверений в ядре Linux для доступа различных приложений к различным сетевым службам, используя одни и те же ключи доступа.
- 7 Реализация загрузки исполняемого модуля из памяти. Работа направлена на изучение механизмов загрузки динамически загружаемых библиотек. В работе необходимо исследовать особенности процесса загрузки модулей

исполняемого кода из динамических библиотек, описать и реализовать упрощённый вариант этого механизма для модуля размещённого в данных или полученного по сети. В работе также можно рассмотреть альтернативные варианты загрузки исполняемого кода в память. Работа может выполняться как под Linux, так и под Windows.

- 8 Реализация загрузки динамической библиотеки в формате PE под Linux. Цель работы состоит в изучении формата запускаемых файлов Windows и механизмов размещения и запуска загруженного кода библиотеки в GNU/Linux. Примером применения аналогичной задачи является загрузка под Linux видео кодеков для Windows.
- 9 Сборка пакета со специализированным ядром в операционной системе GNU/Linux. Целью работы является исследование процесса конфигурирования, сборки и ядра операционной системы под задачу (уменьшение объёма занимаемого пространства в памяти, поддержка расширенного режима адресации памяти 32-битном режиме работы,), а также упаковки результат в виде тиражируемого решения.
- 10 Реализация альтернативного механизма входа в систему Windows. Работа направлена на исследование механизмов локальной аутентификации и авторизации в семействе ОС Windows NT, основанной на использовании расширяемого набора модулей, используемых через интерфейс библиотеки установки контекста для запускаемых процессов GINA.
- 11 Исследование современных реализаций системы запуска ОС GNU/Linux. В работе предполагается разобрать традиционный подход к загрузке ОС SystemV, провести его сравнение с современными альтернативами (upstart, initng, systemd) и дать оценку рассмотренным альтернативам в соответствии с теми задачами, для которых применение этих альтернатив оправдано.
- 12 Исследование специализированных возможностей ядра Linux. Работа направлена на исследование специализированных возможностей предоставляемых ядром Linux, но редко используемых традиционных в приложениях с силу кроссплатформенности и совместимости, которой придерживаются разработчики приложений и GNU/Linux дистрибутивов. Предполагается рассмотреть системный вызов создания контекста исполнения clone(), средства управления привилегиями (capabilities), средство группового управления ресурсам процессов (cgroups), виртуализатор уровня ядра, использующий механизмы аппаратной поддержки виртуализации (KVM), модуль поддержки запуска сторонних форматов исполняемых файлов (binfmt_misc), расширяемых механизм уведомлений о событиях (epoll), быстрый механизм синхронизации процессов (futex — fast mutex), модуль создания файловой системы в пространстве пользователя в виде службы

(FUSE), расширенный механизм контроля доступа (SELinux), Механизм быстрой перезагрузки ядра (kexec) и т.д.

- 13 *Исследование специализированных возможностей ОС OpenSolaris. Работа направлена на исследование специализированных возможностей предоставляемых ОС OpenSolaris. Предполагается рассмотреть файловую систему ZFS, предоставляющую широкие возможности для резервного копирования, распределения дисковых ресурсов и управления их состоянием, встроенный механизм динамической трассировки приложений уровня ядра ОС (DTrace), технология виртуализации на уровне операционной системы (Solaris Containers), механизм создания множества виртуальных сетевых интерфейсов (Crossbow) и т.д.*
- 14 *Исследование специализированных возможностей ОС FreeBSD. Работа направлена на исследование специализированных возможностей предоставляемых ОС FreeBSD. Модули совместимости с приложениями, предназначенным для выполнения в других ОС (включая Linux, SCO UNIX, NetBSD и BSD/OS), поддержка файловой системы ZFS, предоставляющей широкие возможности для резервного копирования, распределения дисковых ресурсов и управления их состоянием, поддержка встроенного механизма динамической трассировки приложений уровня ядра ОС (DTrace), механизм виртуализации уровня операционной системы (Jail) и т.д.*
- 15 *Исследование расширяемого микропрограммного интерфейса EFI. Работа направлена на исследование интерфейса между операционной системой и микропрограммами (EFI — Extensible Firmware Interface), управляющими низкоуровневыми функциями оборудования, предназначенного корректно инициализировать оборудование при включении системы и передать управление загрузчику операционной системы. EFI предназначен на замену BIOS — традиционного интерфейса используемого всеми IBM PC-совместимыми персональными компьютерами.*
- 16 *Современный формат размещения таблиц разделов GPT. Работа направлена на исследование нового формата таблицы разделов (GUID Partition Table), являющегося частью расширяемого микропрограммного интерфейса (EFI). Новый формат таблицы разделов предназначен в качестве замены главной загрузочной записи (MBR).*
- 17 *Программные интерфейсы и режимы работы современных графических адаптеров. Работа направлена на исследование программных интерфейсов современных графических адаптеров, реализуемых в драйверах ОС, и режимов работы, определяющих отображение памяти графических адаптеров в адресное пространство управляющих модулей ядра и приложений.*

- 18 Двоичный интерфейс приложений x32-psABI для архитектуры x86_64 в ОС GNU/Linux. Работа направлена на исследование реализации режима поддержки двоичного интерфейса x32-psABI в ядре Linux и базовых библиотеках ОС, в котором используются практически все преимущества x86_64, но предлагает 32-битный размер указателей, что позволяет сократить объёмам используемой оперативной памяти в два раза.
- 19 Реализация Wine-сервера в ядре Linux. Исследование решения по переносу реализации службы поддержки среды для запуска Windows-приложений в Unix-подобных операционных системах в ядро Linux. Реализация службы wine-сервер позволяет уменьшить накладные расходы на трансляцию системных вызовов, за счёт уменьшения числа переключений контекста процессов, и ускорить работу запускаемых приложений в несколько раз. В результате, некоторые приложения смогут работать быстрее, чем в родной операционной системе.
- 20 Архитектура улучшенного программируемого контроллера прерываний APIC. Работа направлена на исследование возможностей предоставляемых улучшенным программируемым контроллером прерываний (Advanced Programmable Interrupt Controller) для многопроцессорных систем, а также изучение работы контроллера на примере модулей использующих его ОС.
- 21 Реализация архитектурно-зависимого приложения в ОС Android. Работа направлена на исследование интерфейса для реализация «родных» приложений (NDK — native development kit) в ОС Android, с использованием. Основной целью использования этого интерфейса, обычно, является реализация высокоэффективного кода игровых приложений, способного запускаться на современных мобильных телефонах, планшетных компьютерах, нетбуках и смартбуках.
- 22 Объектная модель ядра Linux и её представление в файловой системе /sys. Работа направлена на исследование объектной модели используемой для организации внутренней структуры ядра и её динамического отображения в пространство пользователя через файловую систему /sys. Результатом работы может быть набор модулей ядра, дающих представление об использовании структур ядра в драйверах устройств.
- 23 Установка GNU/Linux с корневой файловой системой JFFS2. Работа направлена на изучение файловой системы для флеш-накопителей JFFS2, и возможностям её использования в качестве корневой файловой системы для ОС GNU/Linux.
- 24 Алгоритм вычисления списка пакетов по замыканию дерева сборочных зависимостей для заданного множества пакетов. Работа направлена на исследование системы установки ПО в ОС GNU/Linux, применяющих

пакетные менеджеры с зависимостями. Целью работы является составление эффективного алгоритма по вычислению минимального списка пакетов необходимых для пересборки заданного множества.

- 25 *Конфликты компоновки динамически загружаемых библиотек на уровне двоичных интерфейсов. В работе рассматривается проблема загрузки в память одной и той же библиотеки разных версий с разными бинарными интерфейсами. Проблема, в основном, встречается при не явной загрузке модулей расширения скомпонованных с разными версиями библиотек. Целью работы является описание причин, определение потенциальных последствий, проблемы и варианты её избежания.*