Linux Kernel

Trainer: Leonid Karpov

Организация курса

Карпов Леонид
leonid.karpov@auriga.com

Структура курса:

1. История и обзор ОС Linux - сегодня
2. Организация файловой системы, системные вызовы
3. Менеджмент памяти
4. Процессы, планировщик задач
5. Обработчики прерываний, примитивы синхронизации

История Linux

История Linux

- Ранние ЭВМ (Z1, ENIAC, Colossus, 1936-1944)
  - Программа задается в аналоговом виде (коммутационная доска, перфокарты)
  - Однотипные задачи
- Развитие ЭВМ, возможность хранить программу в ОЗУ (Manchester Mark I, EDSAC, EDSAC-2, IBM 709, 1945-1960)
  - Выполнение программы занимает меньше времени, чем её запуск
  - Появляется необходимость планировать задачи и разделять ресурсы между пользователями
  - Batch processing – прообраз скриптов, выполяется программой-монитором (предшественник ОС)
- Появление первых ОС:
  - CTSS (Compatible Time-Sharing System, 1961, MIT) - многопользовательская ОС, time-sharing, защищенный режим ядра, консольные команды
  - Multics (Multiplexed Information and Computing Service, 1964, MIT, GE, Bell Labs (AT&T)) - концепция виртуальной памяти, иерархическая файловая система, динамическая линковка

История Linux

- UNIX, изначально Unics (Uniplexed Information and Computing Service, Bell Labs (AT&T), 1969-1978)
  - Многопользовательная многозадачная система
  - Терминал и командная строка (Bourne Shell)
  - Представление устройств в виде файлов
  - Мультиплатформенность
- BSD (Berkeley Software Distribution, 1978)
  - Изначально – аддон для Research UNIX, затем код UNIX был заменён
  - Лицензия BSD
  - Сетевой стек, сокеты Беркли (POSIX)
- Коммерческие версии UNIX
  - UNIX System V (начиная с 1983, AT&T)
  - SunOS (ныне Solaris)
  - IBM AIX
  - HP-UX
  - Начало разработки единого стандарта Single UNIX Specification

История Linux

- MINIX (Andrew Tanenbaum, 1987)
  - UNIX-подобная ОС, не наследник UNIX
  - Микроядерная архитектура
  - Первые версии - пример реализации архитектуры в учебных целях
- Проект GNU (GNU’s not Unix, Richard Stallman, 1983)
  - Идея свободного ПО с открытым исходным кодом (GPL)
  - Hurd – официальное ядро GNU OS
  - К 1991 г. разработано множество утилит (emacs, bash, gcc etc.)
- Linux (Linus Torvalds, 1991)
  - Minix и GNU оказали влияние на ОС
  - Была задумана совместимой с UNIX
  - Цель реализации – воплотить свои идеи реализации ОС, хобби
  - Портированы bash и gcc
  - Разработка подхвачена энтузиастами

История Linux

- Стандарты UNIX:
  - POSIX (Portable Operating System Interface) – набор стандартов, определяющих интерфейс ОС, требует сертификации
  - SUS (Single UNIX Specification) – определяет реализацию интерфейса, включает в себя POSIX, позволяет использовать торговую марку UNIX – требует оплаты
  - Примеры сертифицированных UNIX систем – IBM AIX, HP-UX, Mac OS X, Solaris
- UNIX-подобные ОС:
  - Поведение соответствует концепциям UNIX/Образованы под влиянием UNIX – четкого определения нет
  - Могут быть сертифицированы по стандарту POSIX, (например, QNX Neutrino, Integrity, LynxOS)
- Стандарты GNU/Linux:
  - Не сертифицирована по стандартам SUS и POSIX
  - Большинство версий – POSIX-совместимые (не сертифицированы, но соответствуют стандарту)
  - Проект LSB (Linux Standard Base) – разработан для дистрибутивов GNU/Linux, опирается на существующие спецификации (POSIX, SUS и другие открытые стандарты), расширяя и дополняя их

История Linux

- Масштаб разработки Linux:
  - 1991 – версия 0.01, 10 тыс. строк кода
  - 1994 - версия 1.0, 310 тыс. строк кода
  - 1996 - версия 2.0, 777 тыс. строк кода
  - 2003 - версия 2.6, почти 6 млн строк кода
  - 2008 – совокупный труд оценен в 73 000 человеко-лет
  - 2011 – версия 3.0, 14 млн строк кода
  - 2015 – версия 4.0, 19 млн строк кода
  - 2019 – версия 5.0, 26 млн строк кода
- Дистрибутивы GNU/Linux:
  - Состоит из ядра Linux, библиотек GNU, оконного менеджера, дополнительного ПО, документации и окружения (KDE, Gnome, xfce…)
  - На данный момент существует 276 известных дистрибутивов (по данным distrowatch.com)
  - Коммерческие дистрибутивы (RHEL, SUSE Linux), часть кода ядра Linux – продукт коммерческой разработки

История Linux

- Распространенность GNU/Linux: (по данным gs.statcounter.com и w3techs.com на 2020 г.)
  - Среди десктопных ОС – менее 2%
  - Среди публичных серверных ОС (веб, почтовые, DNS сервера) – около 40%
  - Среди встраиваемых систем – около 38%
  - Среди мэйнфреймов – около 28%
  - Среди суперкомпьютеров – 100% (на 2017 г.)
  - Android (основан на ядре Linux) среди мобильных устройств – около 70%
- Плюсы Linux:
  - Возможность тонкой настройки
  - UNIX-подобность – позволяет легко переключаться между разными UNIX-подобными ОС
  - Широкая база драйверов и поддерживаемых платформ
  - Открытый исходный код – возможность изучения и экспериментирования
  - Множество дистрибутивов

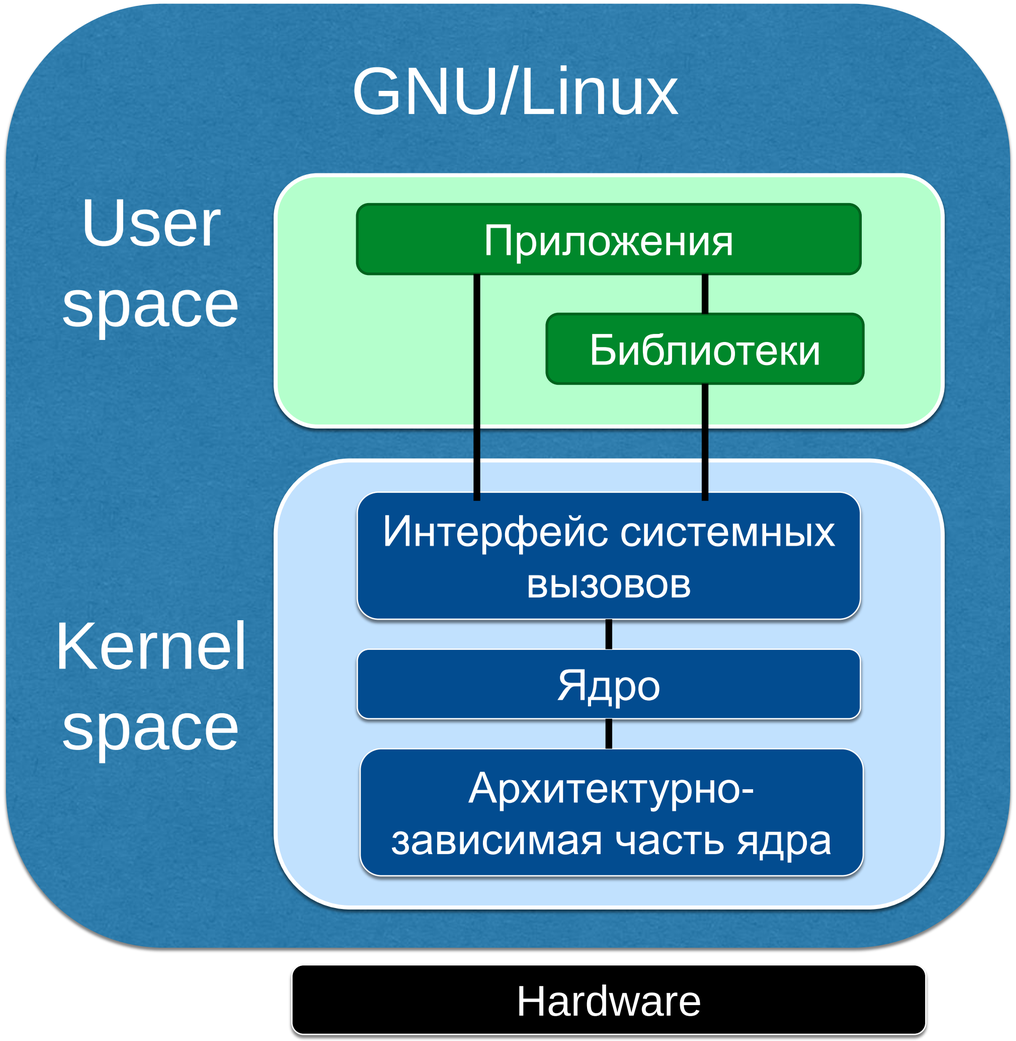
Обзор Linux

Обзор Linux

- Задачи ОС:
  - Абстрагирование от аппаратной платформы – предоставление интерфейса пользовательским программам (системные вызовы, интерфейс файловой системы)
  - Защита системных ресурсов (пространство пользователя и ядра)
  - Распределение системных ресурсов (изолированные процессы, контроль доступа к файлам)

- Основные концепции GNU/Linux:
  - Является многопользовательской системой
  - Предоставляет различные права доступа к файлам на основании User ID и Group ID
  - Является многозадачной системой – поддерживает изолированные друг от друга процессы
  - Воплощает модульную архитектуру ядра ОС
  - Использует интерфейс командной строки
  - Поддерживает простое конфигурирование с помощью текстовых файлов
  - Представляет ресурсы в виде файлов
  - Поддерживает конвейеры
  - Воплощает иерархическую файловую систему

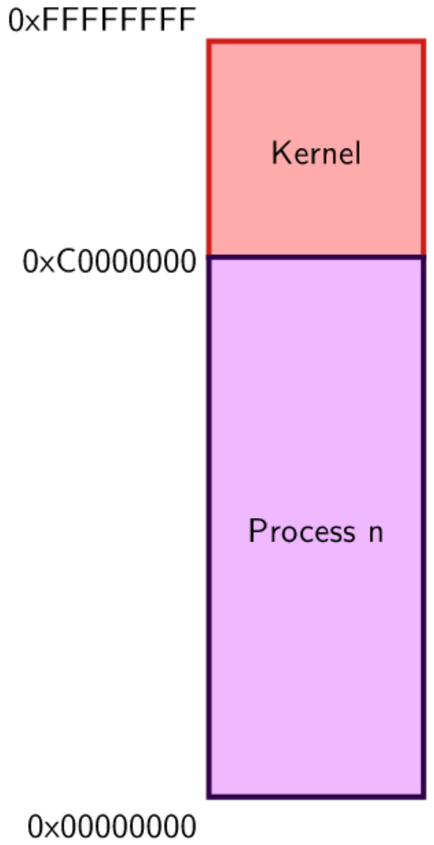
Обзор Linux



- User space:
  - Пользовательский интерфейс (Библиотеки, API)
  - Системные задачи (утилиты, демоны, системные программы)

- Kernel space:
  - Взаимодействие с аппаратной частью (процессор, ввод/вывод, прерывания, прочие устройства)
  - Менеджмент ресурсов (процессы, планировщик, отображение памяти)

Обзор Linux



- Виртуальное адресное пространство – это максимально доступное адресное пространство для отображения физической памяти
- Объем виртуальной памяти и объём физической оперативной памяти не связаны друг с другом

- Адресное пространство включает:
  - Kernel space – располагается по смещению PAGE_OFFSET, адреса отображены на одинаковые физические страницы для всех процессов,
  - User space – уникально для процесса, адреса отображены на разные физические страницы (хотя виртуальные адреса могут быть одинаковыми)
  - Смещение и размер обоих пространств зависит от архитектуры и конфигурации системы – разрядности CPU, конфигурации Page Table
  - Смещение пространства ядра может быть сконфигурировано случайным образом (CONFIG_RANDOMIZE_BASE)

Обзор Linux

Основные состояния ОС:

- Режим пользователя:
  - Непривилегированный режим процессора (напр., Ring 3 для x86, PL0 для ARM)
  - Доступ к памяти процесса в User Space, контролируется с помощью Page Table
  - Ограниченный набор инструкций процессора
  - Нет прямого доступа к Hardware – для этого нужны драйвера
  - Запросы к ОС – через системные вызовы
- Режим ядра:
  - Привилегированный режим процессора (напр., Ring 0 для x86, PL1 для ARM)
  - Доступ ко всей памяти
  - Исполнение привилегированных инструкций
  - Прямой доступ к HW

Обзор Linux

Способы исполнения программ:

- Процесс:
  - Создается как потомок уже существующего процесса
  - Имеет собственное пространство пользователя
  - Может создать несколько потоков исполнения
- Поток:
  - Использует адресное пространство процесса
  - Может переключаться между режимами пользователя и ядра с помощью прерываний или системных вызовов
  - Собственный user stack в user space
  - Собственный kernel stack в kernel space
- Ядерный поток:
  - Выполняется только в режиме ядра
  - Не имеет пользовательского пространства
  - Обычно создаются для выполнения регулярных действий в ядре (напр. обработка прерываний от устройства или регулярные действия)

Вопросы