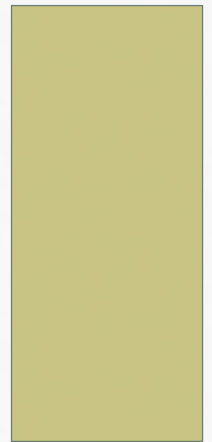


ОБО ВСЕМ И ПРО UNIX



ПРЕПОДАВАТЕЛЬ И ССЫЛКИ

- Кузнецов Антон Михайлович
- +7(921) 310-96-77
- anton.m.kuznetsov@gmail.com
- <http://mit.spbau.ru/sewiki>

ТРЕБОВАНИЯ К ЗАЧЕТУ

- Сдать домашние задания на необходимое количество баллов.
- Написать самостоятельные работы
- ...

ИСТОРИЯ UNIX



ИСТОРИЯ UNIX

1960е: Multics (Multiplexed Information and Computing Service) (MIT, Bell Labs, GE) - многопроцессорность, модульность, для мейнфреймов

1969: Unics (UNIpIplexed Information and Computing System) (Ken Thompson, Dennis Ritchie) – time-sharing system, иерархическая файловая система, процессы, файлы устройств, командная строка.

1971: Unix – работа с текстом (форматирование, редактор), написана на ассемблере.

1973: 3 версия, переписан на языке C (все кроме ядра и I/O).

1975: 4-6 версии – pipes.

1979: 7 версия, общий идеологический предок большинства существующих операционных систем.

ИСТОРИЯ LINUX

1978: BSD UNIX (основана на Unix v6) – университет Беркли.
(сеть TCP/IP)

1982: UNIX System III – коммерческая версия от AT&T (Bell Labs)
(сеть TLI)

1980е: BSD - NFS, VFS, Ethernet; System V - vi, RFS

1983: GNU (Richard Stallman) - идея создать свободный Unix с нуля, gcc.

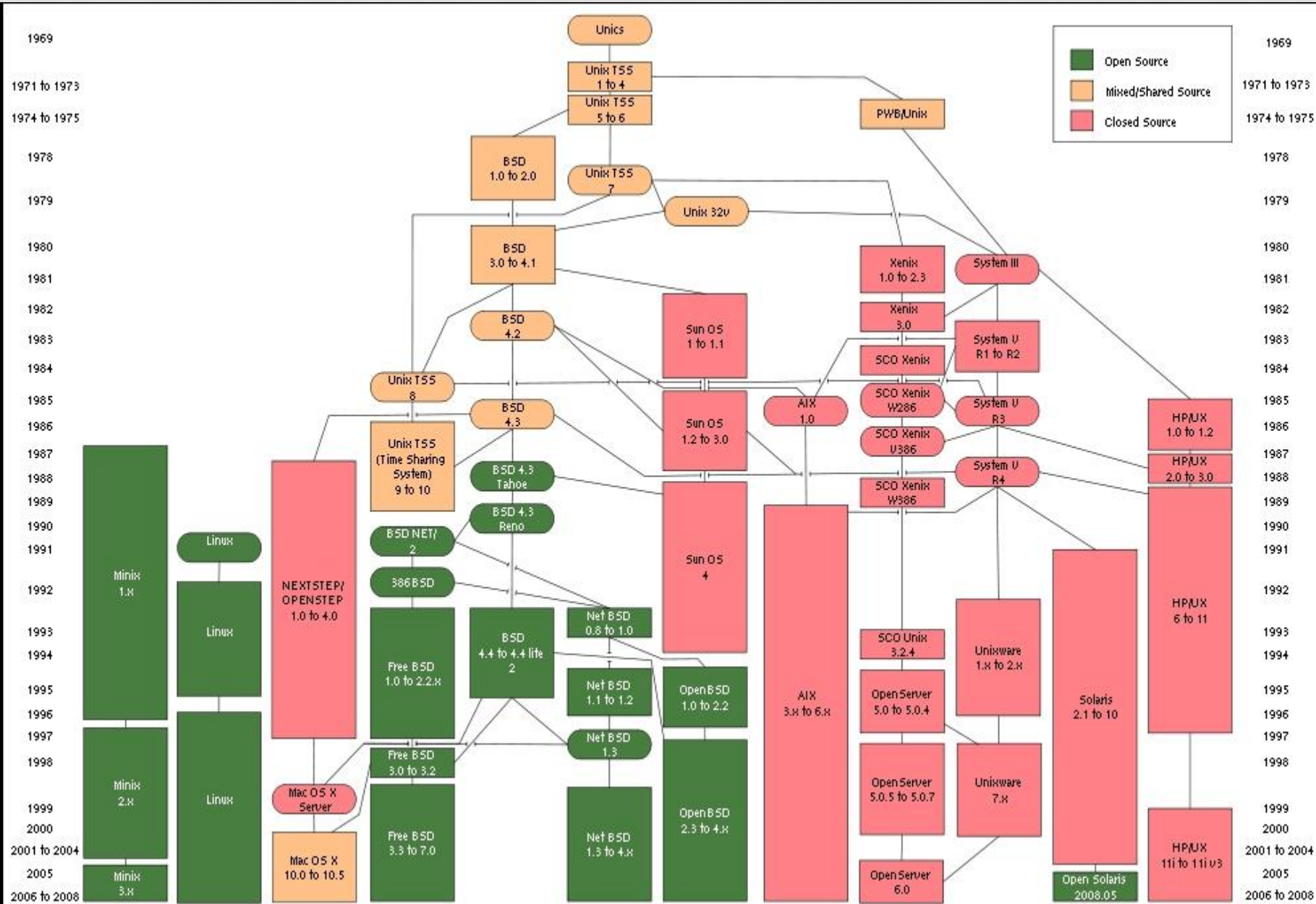
1985: Free Software Foundation, GNU GPL.

1987: Minix (Andrew S. Tanenbaum).

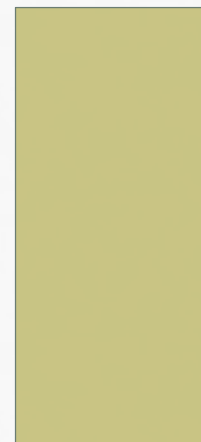
1991: Linux (Linus Torvalds).

Другие Unix-like системы:

AIX (IMB), Solaris (Oracle), HP/UX (HP), Mac OS X (Apple),
OpenServer (SCO), FreeBSD, OpenBSD, NetBSD.



КРАТКИЙ ОБЗОР



ОБЗОР

Состоит из:

- **Kernel** - ядро, основная программа Unix. Контролирует железо, CPU, память, диски, сеть и т.п.
- **Shell** - интерфейс между пользователем и ядром. Интерпретирует вводимые команды и передает их ядру.
- Много пользователей могут использовать одну машину одновременно (Multi-user & Multi-process).
- Данные, директории, процессы, жесткий диск и практически все – файлы.
- Процесс – работающая программа, идентифицируемая уникальным pid.
- Зависимость от регистра (case sensitivity)!

КРАТКО ПРО ДИРЕКТОРИИ

- Файлы размещаются в директориях.
- Директории складываются в иерархическую структуру.
- Можно добавлять или удалять любую директорию в/из дерева директорий.
- Корневая (root) директория "/".
- У каждого пользователя есть своя домашняя (home) директория.

СТАНДАРТНЫЕ ДИРЕКТОРИИ

- **/bin** - файлы, необходимые для корректной работы системы, доступны для использования всеми пользователями.
- **/sbin** - то же самое, но для супер-пользователя только.
- **/dev** - устройства (например, жесткие диски, DVD-ROM).
- **/etc** - различные конфигурационные файлы системы.
- **/home** - там размещаются все домашние директории.
- **/var** - файлы, которые меняются часто, и в которые можно писать.
- **/tmp** - временные файлы.
- **/usr** - все установленные пакеты программ, документация, исходный код ядра и система X Window
- **/proc** - различная информация, которую ядро может сообщить пользователям

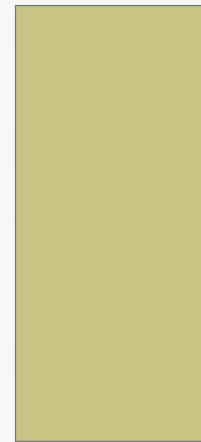
ПОЛЬЗОВАТЕЛИ И ПРАВА

- Супер-пользователь (root, superuser) - единственный на систему, администратор.
- Остальные – нормальные пользователи.
- Права доступа устанавливаются для трех сущностей: пользователя, группы и всех остальных

UNIX PHILOSOPHY

- Write programs that do one thing and do it well.
- Write programs to work together.
- Write programs to handle text streams, because that is a universal interface.

ВВЕДЕНИЕ В КОМАНДЫ UNIX



UNIX SHELL

UNIX Shell - командный интерпретатор, используемый в операционных системах семейства UNIX, посредник между пользователем и операционной системой.

- Пользователь может либо давать команды операционной системе по отдельности, либо запускать скрипты, состоящие из списка команд.

POSIX

POSIX (ISO/IEC 9945) – (Portable Operating System Interface for Unix - Переносимый интерфейс операционных систем Unix) — набор стандартов, описывающих интерфейсы между операционной системой и прикладной программой.

POSIX

Том 3. оболочка и утилиты

- Определён язык оболочки, включающий конструкции последовательного (перевод строки, точка с запятой), условного (if, case, ||, &&) и циклического (for, for in, while, until) исполнения команд, а также оператор присваивания.
- Стандартом также определён режим редактирования вводимых команд, являющийся подмножеством команд стандартного текстового редактора (vi).

UNIX SHELLS

- **Bourne shell compatible**

sh, ash, bash, dash, ksh, zsh...

- **C shell compatible**

csh, tcsh

- **Exotic**

es (functional), psh (Perl), pysh (Python)

BOURNE SHELL

- Разработана Стивеном Борном из Bell Labs и выпущенная в составе 7-го издания операционной системы UNIX (UNIX Version 7, 1977).
- Основана на языке ALGOL 68
- "Nobody really knows what the Bourne shell's grammar is. Even examination of the source code is little help."
 - *Tom Duff*

C SHELL

- Разработан Биллом Джоем (университет Беркли) в конце 1970х
- Основан на языке C.
- Имеет множество улучшений по сравнению с sh

KORN SHELL

- Разработан Дэвидом Корном в начале 1980х. Имеет полную обратную совместимость с Bourne shell и включает в себя возможности C shell.
- «The POSIX standard specifies its standard shell as a strict subset of the Korn shell»

BOURNE AGAIN SHELL (BASH)

- Разработан для проекта GNU в замен sh Брайаном Фоксом и представленный в 1989 году. Собрал в себе множество идей из csh и ksh.
- Основной shell для Linux и MacOS

BASH FEATURES

- Редактирование командной строки
- История команд
- Стек каталогов
- Переменные \$RANDOM и \$PPID
- Автозавершение
- ...

ВНУТРЕННИЕ КОМАНДЫ

- **echo** - выводит выражение или содержимое переменной (на stdout)
- **printf** - команда форматированного вывода, расширенный вариант команды echo
- **read** - «читает» значение переменной со стандартного ввода (stdin), в интерактивном режиме это клавиатура

HELLO WORLD!

- `echo "Hello World!"`
- `printf "%s\n" "Hello World"`

КЛЮЧИ

- Любая команда может быть запущена с ключами:
- **\$ echo -e "a\n"**
- a

- -h, --help – помощь
- -? – помощь (работает не у всех команд)

- Группировка ключей:
- **ls -ltra**

ПОМОЩЬ

- **help COMMAND** – выводит краткую справку по использованию внутренней команды *COMMAND*
- **man COMMAND** – выводит подробную справку по использованию *COMMAND*
- **info COMMAND** – дополнительная информация о команде

РАБОТА С ФАЙЛАМИ

- `cd` – перемещение по директориям
- `pwd` – текущий каталог
- `mkdir` – создание каталогов
- `rm` – удаление файлов и каталогов
- `cp` – копирование файлов
- `touch` – создать файл
- `ls` – вывод содержимого каталога

ПОЛЕЗНЫЕ КОМАНДЫ

- more, less
- head, tail
- wc
- sort
- tee
- find
- locate
- grep

АВТОЗАВЕРШЕНИЕ

- Напечатать первые буква команды
- Нажать Tab
- Если один вариант – автоматическая печать команды
- Если нет – нажав еще раз Tab увидим полный список вариантов

ПОТОКИ

- Поток номер 0 (stdin) – входные данные.
- Можно изменять цель с помощью символа >
- **grep 'apt' < a**
- Поток номер 1 (stdout) - выходные данные
- Можно перенаправить символом >
- **ls > a.txt**
- **grep 'apt' < a > b**
- Поток номер 2 (stderr) - ошибки
- Перенаправлять символом 2>
- **ping -c 5 google.com > result.txt 2> error.txt**

ПОТОКИ

- > - перезапись
- >> - добавление в конец

КОНВЕЙЕР (PIPE)

- Конвейер передает вывод предыдущей команды на ввод следующей или на вход командного интерпретатора. Метод часто используется для связывания последовательности команд в единую цепочку. Конвейер обозначается следующим символом: |.
- **cat filename | grep something**

HOTKEYS

- Ctrl-C - посылает сигнал SIGINT, заканчивает процесс.
- Ctrl-D - завершение ввода с клавиатуры (EOF).
- Ctrl-Z - посылает сигнал SIGTSTP, приостанавливает процесс.

ЗАДАНИЕ

- Написать команду, которая в текущей директории считает кол-во файлов с расширением .txt.
- Команда cal. Используя её, вывести текущее число.
- В файле ip.txt записан IP адрес. Пропинговать компьютер и записать результат в res.txt, а ошибки — в err.txt.
- Создать файл, хранящий в себе список процессов, которые начинаются на r.
- Напишите команду, которая копирует содержимое файла в тот же файл.