Введение

Сейчас трудно найти человека, который бы ни разу не пользовался компьютером, мобильным телефоном, планшетом или другими гаджетами. На каждом из них установлена операционная система.

Операционная система — это комплекс программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем. Операционная система – это основа, без которой невозможно работать ни с одной программой на компьютере. ОС обеспечивает работу устройств, которые подключают к компьютеру: мышки, колонок, клавиатуры, роутера. Связывает с устройствами программы и приложения, которые запускают на компьютере. Например, аудиоплеер — программа — передает звук в физические колонки благодаря ОС. Запускает, устанавливает и удаляет программы, приложения, игры. Создает для пользователей интерфейс: рабочий стол, папки, ярлыки, окна. ОС бывают клиентские и сетевые, однозадачные и многозадачные, однопользовательские и многопользовательские. Популярные ОС: для десктопа — Windows, MacOS, Linux, для мобильных устройств — iOS, Android.

Актуальность данного проекта связана с тем, что взаимосвязь компьютерных наук с каждым днём становится теснее, а развитие операционных систем приобретает все большие масштабы. Поэтому для того, чтобы идти в ногу со временем необходимо понимать как же работает операционная система изнутри.

Я увлекаюсь программированием, хочу связать свое будущее с разработкой программного обеспечения и поддержкой Linux серверов. Следовательно, решила изучить операционные системы более детально. Мне необходимо понимать, как работает операционная система изнутри, чтобы в дальнейшем разработка моих проектов была более успешной.

*Цель проекта:* создать операционную систему JellyOS с нуля.

*Задачи проекта:* изучить возможности Linux систем; изучить основные этапы разработки операционной системы; ознакомиться с возможностями языка программирования C/C++; понять работу ядра Linux.

*Актуальность проекта* заключается в следующем: развитие операционных систем приобретает все большие масштабы. Поэтому для того, чтобы идти в ногу со временем необходимо понимать, как же работает операционная система изнутри. Она будет достаточно востребована у определенной ниши людей, а именно у разработчиков ПО, системных администраторов и любителей экспериментировать над ОС, поскольку аналогов крайне мало и они требуют сложной и долгой конфигурации.

1 Основная часть

1.1

[1] Операционные системы можно разделить на несколько основных типов: десктопные, мобильные и серверные.

Десктопные операционные системы рассчитаны на персональные компьютеры и ноутбуки. Примеры: Microsoft Windows, macOS (ранее Mac OS X), Linux (различные дистрибутивы). Предназначены для использования конечными пользователями и обеспечивают интерфейс для запуска прикладных программ.

Мобильные операционные системы разработаны специально для мобильных устройств, таких как смартфоны и планшеты. Примеры: Android, iOS (Apple), HarmonyOS (Huawei), LineageOS и др. Обладают оптимизированным пользовательским интерфейсом и функциями, связанными с мобильными приложениями.

Серверные операционные системы предназначены для установки на серверах и обеспечивают функциональность, необходимую для управления сетевыми ресурсами, хранения данных, обеспечения безопасности и управления сетевой инфраструктурой. Примеры: Windows Server, Ubuntu Server, Debian, Red Hat Enterprise Linux.

Функции операционной системы: руководство загрузкой процессора, управление свободными ресурсами памяти, распределение потоков процессов и поддержка многозадачности (одновременного выполнения нескольких задач).

Операционные системы могут иметь графическую оболочку для комфортного взаимодействия пользователя с техникой. Графический интерфейс — «лицо» каждой ОС, которое позволяет открывать окна, иметь ярлыки на рабочем столе, позволяя неопытным пользователям интуитивно пользоваться компьютером.

Для разработки своей операционной системы я остановилась на ядре Linux Оно имеет существенные отличия от Windows. Главное и ключевое — расположение системных файлов, то есть файловая система. В Windows каждая программа и приложение имеет свою папку, в которой находятся все файлы, отвечающие за их работу. Linux использует другую сортировку: файлы распределяются в зависимости от задачи, которую они выполняют. А также Linux обладает следующими положительными возможностями. Система бесплатна: ее можно свободно скачать с сайта желанного дистрибутива. Бесплатно и все программное обеспечение. Программы можно модифицировать при наличии знаний в программировании. Безопасность с точки зрения проникновения в систему вирусов на очень высоком уровне, поскольку вирусы для Linux крайне сложно написать и их легко обнаружить самостоятельно. Возможность запуска приложений для Windows с помощью эмуляторов. А также один из ключевых моментов — это чрезвычайно высокая оптимизация и скорость работы системы, что способствует ее стабильности. Отрицательные моменты в данной ОС присутствуют. Есть проблемы с поддержкой аппаратного обеспечения и внешних устройств (например, принтеров), но с каждым обновлением это улучшается. Программ и игр для Linux создается мало, поскольку разработчики не готовы вкладываться в нишевую аудиторию (2% пользователей Linux используют его для игр). Чтобы работать с Linux, нужно постоянно самообучаться, совершенствовать знания и быть крайне терпеливым.

1.2 Разработка операционной системы «JellyOS»

JellyOS - операционная система для продвинутых пользователей. Уже пару десятков лет существуют системы, подходящие начинающим, поэтому делать еще одну такую ​​очередную нет смысла. В составе этой системы уже готовы утилиты для управления серверами, нетворкинга, кибербезопасности, игр. Таким образом, это достаточно универсальная система, обеспечивающая более низкую степень смягчения круга людей, таких как я.

Для начала была осуществлена подготовка хост-системы (Gentoo Linux): раздел диска и его форматирование (системный раздел EFI и для файловой системы), создание временного нового пользователя для новой ОС ("lfs"). Затем были созданы новые папки (bin, sbin, usr, sources, lib, lib64, var, tools), компиляция кросс-наборов инструментов (binutils, gcc, linux API, glibc, libstdc++ из gcc), а также кросс-компиляция временных утилит (m4, ncurses, bash, coreutils, diffutils, file, gawk, grep, gzip, make, patch, sed, tar, xz) и переход в окружение chroot с установкой дополнительных временных утилит (gettext, bison, perl, python, texinfo, util-linux). После прохождения этапов новая ОС практически уже самостоятельна и команды вводятся в терминал полностью от лица нового пользователя. Установила дополнительные программы для полноценного использования системы (man pages, iana-etc, glibc, zlib, bzip, xz, zstd, file, readline, m4, bc, flex, tcl, expect, dejaGNU, pkgconf, binutils, gmp, mpfr, mpc, attr, acl, libcap, libxcrypt, shadow, gcc13, ncurses, sed, psmisc, gettext, bison, grep, bash, libtool, gdbm, gperf, expat, inetutils, less, perl, xml::parser, intltool, autoconf, automake, openssl, kmod, libelf из elfutils, libffi, python3, flit-core, wheel, setuptools, ninja, meson, coreutils, check, diffutils, gawk, findutils, groff, GRUB, gzip, IProute, kbd, libpipeline, make, patch, tar, texinfo, vim, markupsafe, jinja, udev из systemd, man-DB, procps-ng, util-linux, e2fsprogs, sysklogd, sysvinit). Для ее запуска системы будет использоваться System V. Далеее установка checkfs, cleanfs, console, functions, halt, ifdown, ifup, localnet, modules, mountfs, mountvirtfs, network, rc, reboot, sendsignals, setclock, ipv4-static, swap, sysctl, sysklogd, template, udev, udev\_retry. Добавлена реализация udev для управления устройствами и модулями ядра. Дополнила ОС параметрами интернет-соединения: создание файлов конфигурации для интернет-устройств (/etc/sysconfig/ifconfig.eth0), настройка DNS и IP-адресов в /etc/resolv.conf, определение имени компьютера по которому он будет определяться в сети (etc/hostname), создание файла /etc/hosts с FQDN (полное доменное имя, англ. fully-qualified domain name) и именем компьютера.

Конфигурация System V. Она будет включать: конфигурация Sysvinit (инициализация ядра через init в файле /etc/inittab), конфигурация системных часов (/etc/sysconfig/lock), настройка консоли (раскладка клавиш, язык, необычные символы, шрифт, юникод и тд.), создание файла rc.site (настройки запуска System V) и его кастомизация, конфигурация языка системы (британский английский), создание файла /etc/inputrc (файл конфигурации для readline, позволяющий менять введенный текст в терминале), создание файла /etc/shells (список командных оболочек: sh, bash). Также настройка запуска непосредственно новой системы: создание файла /etc/fstab (список файлов систем и место их установки в файловой системы), установка и настройка Linux-6.7.4, загрузка ядра. В качестве загрузчика будет использоваться GRUB. Создание /boot/grub/grub.cfg (файл конфигурации GRUB). Ввод информации о версии системы в /etc/os-release и запуск системы.



Рисунок 1 – Рабочий стол JellyOS



Рисунок 2 – Окно с терминалом



Рисунок 3 – Лаунчер и поиск приложений и файлов



Рисунок 4 – Работа с браузером Firefox, пример реализации многозадачности

Реализован в проекте следующий софт**:**

* Менеджер окон Hyprland
* Строка состояния Waybar
* Лаунчер (поиск) приложений rofi-wayland
* Скриншоты grim, slurp, swappy
* Терминал kitty
* Браузер firefox
* Текстовый редактор vim
* Уведомления swaync
* Плеер медиа mpv
* Блокировка экрана hyprlock
* Настройки GTK nwg-lock
* Настройка приложений Qt qt5ct, kvantum
* Генератор палитр цветов wallust
* Polkit - polkit-gnome
* Выход из учетной записи, меню включения и перезагрузки wlogout
* Манипуляция изображениями imagemagick
* Регулировка яркости монитора Brightnessctl
* Командное заключение zsh + oh-my-zsh
* Блютуз bluez, blueman
* Изображение при открытии терминала pokemon-colorscripts
* Просмотр изображения eye of gnome
* Монитор ресурсов btop++

Литература

1. <https://blog.skillfactory.ru/glossary/operaczionnaya-sistema/>

2. <https://www.linuxfromscratch.org/lfs/>

3. <https://www.linuxfromscratch.org/blfs/>

4. <https://wiki.archlinux.org/title/Main_page>

5. <https://wiki.gentoo.org/wiki/Main_Page>