1.Ярдо \*nix:

Ядро - набор подсистем и программ, управляющие оборудованием и программами пользователя. В ядре выполняются служебные операции ОС. Взаимодействие программ пользователя происходит с помощью интерфейса системных вызовов.

Интерфейс системных вызовов - стандартизированное средство взаимодействия программы пользователя и ядра: доступ к сервисам ядра можно получить только через этот интерфейс.

Ядро системы разработано таким образом, что его легко можно приспособить практически под любой микропроцессор.

Основные подсистемы:

Управление памятью

Управление процессами и потоками

Виртуальная файловая система

Платформозависимый код

Драйвера устройств

2.Фамлова система:

В Unix все является файлом кроме потоков, процессов и ядра.

7.фунлцоалне элементе ЭВМ

Триггер/регистр/элементе хранена(dram-sram)/вентиле(x & 0)/сумматор/тактовой генератор

8.архитектура ЭВМ

Гаравадская:память инструкции и данный не в то же место это позволит оптимизировать

Фон неймна: они вместе

9.адресумая память

Diagram, schematic

Description automatically generated

10:алк куммотатор блок признак резултатов

Diagram

Description automatically generated

11:форматы команды

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

15:Архатиктура систмм в много процессых систима

Uma:мулты проциссор 1 памят

1:все процессоры иммет равны Тд ---- 2:каждый процессор иммет равны возможности запис,читне

Numa: мулты проциссор и памят ---- 1: процессоры иммет рпзный Тд

16:архитектра multi-процессоров risc sisc

Risc: болше регистри – выполнет инструкце в оден такт ---обем код болое

Cisc:меншее регистр -– выполнет инструкце в болше оден такт--- обем код менше

Характарске памят

Месторасположение – процессорные, внутренние, внешние

● Емкость – В метрических (Кило-) и двоичных (Киби-) множителях

● Единица пересылки – Слово, строка кэша, блок на диске

● Метод доступа – Произвольный (адресный), ориентированных на записи (прямой), последовательный, ассоциативный

\* Быстродействие и временные соотношения

\*стоимост

Асирхоная памят

Diagram

Description automatically generated

Синхроная памят :

Address -> ~~ras~~ -> ~~cas~~ -> ~~outputenable~~ -> read/write -> databus

17: Память, с последовательным доступом

ленточных накопителях. ленточные накопители медленные, но цена за единицу объема памяти мала, поэтому на них хранят огромные массивы данных.

Скорость доступа определяется скоростью прокрутки этой самой ленты, но механика не позволяет сделать эту скорость очень высокой Поэтому

современные ленточные накопители работают по принципу наклонно-строчной магнитной записи Лента проходит под вращающейся головкой под некоторым углом, обеспечивая бОльшую плотность записи и скорость

18.кэш памят

В ассоциативной памяти: мы достаем нужные данные по их содержанию, а не их местоположению. Для этого каждая ячейка имеет в себе компаратор, которые может обнаружить соответствие. Найденная ячейка попадает в регистр совпадений и далее на комбинационную схему. Такая память быстрее чем обычная RAM, но и стоит дороже за счёт компараторов и более трудной логики. Принцип ассоциативной памяти использует кэш.

19.перамида памят

Line chart

Description automatically generated with medium confidence

21:сеть ЗВМ:

Данные разбиваются на пакеты, которые последовательно передаются. Это сделано для того, чтобы один процесс передачи данных не занимал слишком много времени.

Концевик - хэш(раньше этот хэш собирался так: мы преобразуем наши

данные в байты и складываем их). Нужен он для проверки корректности передачи данных

Классификация компьютерных сетей:

1. Сети можно классифицировать по размеру:PLMW
2. По принадлежности:
   1. Офисные/Корпоративные/ Частные VPN
3. По назначению: Вычислительные/ Информационные/И-В/И-УПРА
4. По области применения:
   1. SAN - сети хранения данных/Серверные фермы.

22.модель взаимодействую открытых систем OSI

остоит из нескольких уровней

Наше сообщение бьётся на пакеты и на каждом уровне они оборачивается в заголовок уровня).

23:ethernet

Chart

Description automatically generated with low confidence

24:систоеве уровен ip,марищрзаця

Diagram

Description automatically generated

25:tcp/udp

TCP — Transmission Control Protocol

– Надежный, управляет перепосылкой данных

– Организует виртуальное соединение между гнездами (Socket=IP:port) на двух системах –

HTTP, FTP, SSH, SMTP, ...

● UDP — User Datagramm Protocol

– Послал сообщение и забыл

– Контроль надежности оставлен разработику

– Максимальная скорость передачи – SNMP, TFTP, DHCP, DNS, ...

