- 源文件.c 预处理-> .i 生成汇编代码-> .s 汇编 -> .o 链接 ->a.out
- New 和 malloc
    - 内存：new在自由存储器静态分配空间，malloc在堆上动态分配
    - 内存大小：new无需指定内存大小(自动分配)，malloc必须指出
    - 分配失败：new分配失败抛出异常，malloc分配失败返回null
    - （大内存数组分配用malloc，保证不会编译报错）
- 链表判断环：hash表或快慢指针
- 进程Process 线程Thread
    - 基本单元：进程是 分配与管理资源的基本单位， 线程是 独立调度的基本单元，常被称为轻量级进程
    - 地址： 进程之间 独立的地址空间，线程 共享本进程的地址空间
    - 一个进程一个ProcessID，一个进程可以有多个线程，同一进程不同线程切换开销小
    - 多进程与多线程，多线程提高的是cpu使用率 而不是运行效率，
- 输入网址到浏览器显示
    - DNS域名解析系统 解析网址->IP地址
    - 客户端向服务端建立 TCP安全连接（三次握手），发起请求
    - 服务器接收请求，进行处理，并响应请求
    - 浏览器接收服务器资源，进行渲染
    - 四次挥手，断开连接
- TCP UDP
    - TCP 传输控制协议
        - 起始：三次握手 SYN->回复ACK，SYN->ACK
        - 传输：一直回复
        - 占用资源多，可靠不会丢包，保证数据顺序
            - Tcp如何保证数据可靠：传输前握手，一直回复机制->丢包重传，Tcp内容中带校验位->防止损失
    - UDP 用户数据报协议
        - 不可靠，可能丢包，不保证数据顺序，占用资源少，传输快
    - 四次挥手：A发送完毕，发FIN->B返回ACK->B接受完发FIN->A返回ACK
- 虚拟地址和物理地址
    - 物理地址：cpu的地址线可以寻址的内存大小，如20根地址线的1MB空间，虚拟内存是进程运行时所有内存空间的总和
    - 虚拟内存地址(32位4G)分页，产生页page，对物理内存地址分页产生页帧page fram

    - 而又存在页表page table，进行这一映射，也是从页号到页帧号的映射。
    - 虚拟内存地址由页号和偏移量组成，先找页号对应的页帧号，如果不在内存内则用页号和偏移量组成物理地址进行访问
- 进程间通信
    - 匿名管道 pipe：半双工，只能具有亲缘关系的进程通信 如父子、兄弟进程
    - 有名管道 FIFO：存在文件路径映射，不局限于亲缘关系
    - 消息队列 message queue: 消息链表
    - 信号量通信：数据操作锁，用于数据操作中的互斥、同步等操作
    - 共享内存通信：当前进程创建能被其它进程访问的一段内存，与信号量通信搭配，效率高
- Http Https
    - http数据未加密，https(相当于ssl+http)安全
    - Http响应快，只用tcp三次握手，https还额外需要ssl的9次处理，共12次
    - https要用到CA证书，需要一定费用，更耗费服务器资源
    - http端口80，https端口443
- 物理 数据链路 网络 传输 会话 表示 应用
- 指针 引用
    - 指针: 存储地址的变量，引用：原变量的别名
    - 指针定义时可以不初始化，可以指向null，之后可以改变，可以多级，必须检查是否为null
    - 引用必须初始化，不可以指向null，不可以改变
- 存储区： 代码区、全局区、常量区、堆、栈
- 全局变量 && 局部变量
    - 全局变量储存在 静态数据区，局部变量储存在堆栈
        - 生命周期是整个函数区间， 声明区间wq
- 堆 栈
    - 堆：完全二叉树(根节点最大或最小)
    - 栈：线性表，先入后出
    - 堆由系统自动分配与释放，栈由开发人员分配或释放
    - 每个进程栈的大小很小
    - 成长方式上：堆向上生长，内存由低到高，栈向下生长，内存由高到低
    - 堆栈溢出->未回收垃圾资源，层次太深的递归调用
- 数组 指针
    - 数组只能在 静态存储区 和栈上被创建，指针可以指向任意内存块
- 虚函数：到达多态效果，保证函数在运行时动态实现，而不是在编译中静态实现
    - 内联函数、构造函数 不能声明虚函数，析构函数通常为虚函数
- 函数重载：同名函数，参数个数、类型、顺序不同

- 多态：不同子类中 同一函数可不同实现；
    - 静态多态(编译时) && 动态多态(运行时)：虚函数，子类和父类的继承
- static关键字
    - 修饰静态全局变量(特点：未经初始化会自动初始化为0，整个文件可见文件之外不可见，在全局数据区)
    - 修饰函数，只在此文件中使用
- const修饰字：修饰常量，只读
- 宏定义：直接替换，不分配内存
    - 预处理的地方将代码展开，不需要额外时间和空间开销（函数调用：保存并记忆现场，来回跳转）
- 内联函数： inline
    - 内联还是函数，很多函数性质：可以重载，相当于函数代码段替换
- volatile：值随时会变，每次都会从变量的地址中读取数据，而不是从寄存器中
- 指针：
    - int *p[10] 有十个指针的数组,指针指向整型数据， = (int *)p[10]
    - int (*p)[10] 指针指向整个数组
    - int *p(int) 函数，参数为整型
    - int (*p)(int) 函数指针，该函数有一个整型参数，返回值为整型
- 大小端：低序存放在起始地址：小端，高序存放在起始地址：大端
- sizeof()
    - 没有成员的结构体：占一个字节
    - char：1，short：2，int：4，long：4，float：4，double：8
    - struct{}要考虑对齐
        - 如包括char、int、float、double，0 0+1->4整数倍为4,4+4->4的整数倍为8,8+4->8的2整数倍为16，最后+8变24，且24位1 4 4 8最小公倍数
        - example
- 二叉树
    - 基本形式
        - 满二叉树：全满，深度k有2^{k-1}个节点
        - 完全二叉树：最底层未满，且为右侧
    - 有序树
        - 二叉搜索树：根节点最大/小
        - 平衡二叉搜索树：空树 或 左右两个子树高度差的绝对值不超过1，且左右两个子树都是平衡二叉树
- 排序算法
    - 直接排序，先找最小，再第二小，依次类推

    - 冒泡排序，O(n^2),空间O(1),左右比较将大的后移，最大的放在最后，之后继续
    - 快速排序，O(nlogn), 空间O(nlogn)，选取基准，大的放右边小的左边，二分思想
    - 归并排序，O(nlogn)，空间0（n），二叉树分而治之
- strlen() 和 sizeof 所占总空间的字节数，后者针对字符数组和字符串，参数必须是字符型指针
    - sizeof() 是一个操作符，返回变量声明收的内存数，strlen是函数
    - strlen是不包括'\0'的长度的，sizeof包括'\0'
- 常见通信接口
    - IIC: Inter-Integrated Circuit 内部内置集成电路总线
        - 半双工 同步 串行
        - 两信号线：串行数据线SDA与时钟线SCL，信号需上拉
            - 起始信号：SCL高，SDA高到低，停止信号：SCL高，SDA低到高,都是主机发送
            - 应答信号：应答位NACK/非应答位NACK，接收端发送
        - 优点：简单，占空间少，多主控
    - UART: Universal Asynchronous Receiver Transmitter通用异步收发器
        - 低速 全双工 异步 串行
        - 双线RX TX通信
        - 缺点：低速，结构复杂
    - SPI： Serial Peripheral Interface 全双工同步主从式通信 串行外设接口
        - 高速 全双工 同步 串行
        - 三信号线：SCLK（串行时钟）、SDI（串行数据输入）、SDO（串行数据输出），加从机选择线CS
        - 一主机多从机，可GPIO软件模拟IIC
        - 缺点：无应答机子
- 操作符
    - <<左移，相当于/2， >>右移 空位补(看系统) >>>右移，空位补0
    - 翻转：与1异或
- 常见的stl容器
    - vector：类数组，地址连续（空间不足，拷贝扩容自动两倍，大对象效率低)
    - list：双向链表，地址不连续
    - deque：双向队列，地址连续，
    - map：底层红黑树(一种二叉查找树，弱平衡二叉树)
        - map内部使用红黑树，默认排序；unordered_map内部使用哈希表，查找效率高

    - set：底层是红黑树
- hash： hash函数->对应关系
    - hash函数冲突：开放定址法、再哈希法、链地址法、建立公共溢出区