|  |  |
| --- | --- |
| **lexicographical** | The program lexicographical order in C language used to arrange a list of strings.  The lexicographical order is also called dictionary order |
| **AMD processor** | Is the AMD Athlon can be classified as a lower-mid-range laptop processor that can  handle the routine computing tasks very well. For heavier tasks recommended are at  least mid-range processors, such as AMD Ryzen or Intel Core i series.  Lenovo IdeaPad 3 15.  2.6 GHz AMD Ryzen 3 5300U processor  8 GB DDR4 memory  256 GB SSD storage  Windows 11 operating system  15.6" FHD screen, 1920 x 1080 resolution  AMD Radeon Graphics  Ports: 1 x USB 2.0, 1 x USB 3.0, 1x USB-C 3.2 Gen 1 Type-C (support data transfer only), 1 x HDMI, 1 x via Combo Headphone/Mic Jack  3-Cell 45 Whr Lithium Polymer battery, up to 12.6 hours  https://nanoreview.net/en/cpu-compare/intel-core-i5-10210u-vs-amd-ryzen-3-5300u |
| **JavaScript** | JavaScript is a **client-side scripting language**, which means the source code is  processed by the client's web browser rather than on the web server. This means  JavaScript functions can run after a webpage has loaded without communicating  with the server. For example, a JavaScript function may check a web form before  it is submitted to make sure all the required fields have been filled out. The Java  Script code can produce an error message before any information is transmitted  to the server.  Like server-side scripting languages, such as php and asp, JavaScript code  can be inserted anywhere within the HTML of a webpage |
| **Cookies** | Cookies are simple text files that contain two pieces of information: a website name  and a unique identifier of some sort. This identifier could be a number or an  alphanumeric string. When you visit a cookie-using website for the first time, it places a cookie in your web  browser. When you visit this website again, it looks for the cookie so that it can tell  whether you have visited before and how to best improve your experience for your  latest visit. For example, the first time you visit a website, it might show you a  message that welcomes you to the site and shows you how to find your way around  via some simple instructions. Now, suppose you go back a couple of weeks later, the  website knows you have been there before (via the cookie in your browser) so it  shows you a 'welcome back' message instead. |
| **sandbox** | sandbox is a mechanism to securely run programs inside an environment from  which it cannot affect other programs and have limited resources to use ;example  web browser ,virtual machine.  What is a Security Sandbox? | Database Doctor |
| **ebpf** | eBPF is a revolutionary technology with origins in the Linux kernel that can run  sandboxed programs in an operating system kernel. It is used to safely and efficiently  extend the capabilities of the kernel without requiring changing kernel source code  or load kernel modules.  Seguridad de red de microservicios basada en eBPF - Code World |
| **Nginx** | Nginx, pronounced as engine-x, is an open-source web server which server more 25%  websites across the globe, and also used as a reverse proxy and load balancer.  Nginx is built to offer low memory usage and high concurrency. Rather than creating  new processes for each web request, NGINX uses an asynchronous, event-driven  approach where requests are handled in a single thread.  Nginx has Worker connections and worker processes, one worker processes can  maintain 1000 worker connections.  Advantages of NGINX:  \* Open source.  \* A high speed web server which can be used as a reverse-proxy server.  \* Can be used better in a virtual private server environment |
| **udev** | Udev is the device manager for the Linux kernel It runs in userspace that creates/  removes device nodes in the /dev directory dynamically. It is the successor of devfs  and hotplug. and the user can change device names using Udev rules. Udev rules  determine how to identify devices and how to assign a name  that is persistent through reboots or disk changes  . |
| VLAN | Virtual LANs (VLANs) allow network administrators to subdivide a physical network into separate  logical broadcast domains. On a standard Layer 2 network, all hosts connected to a switch are  members of the same broadcast domain; and broadcast domains can only be physically  separated across different switches by routers.  VLANs are identified by a VLAN ID (a number between 0 – 4095), with the default VLAN on any  network being VLAN 1. Each port on a switch or router can be assigned to be a member of a  VLAN (i.e., to allow receiving and sending traffic on that  VLAN). For example: on a switch, traffic that is sent to a port that is a member of VLAN 100, may  be forwarded to any other VLAN 100 port on the switch, and it can also travel across a  trunk port (connections between switches) to another switch and forwarded to all VLAN  100 ports on that switch. Traffic won't, however, be forwarded to ports that are on a different  VLAN ID.  As VLANs are a Layer 2 protocol, Layer 3 routing is required to allow communication between  VLANs, in the same way a router would segment and manage traffic between two subnets  on different switches. In order to implement VLANs, the routers and switches must support VLANs. Although there are  several proprietary protocols in existence, the most commonly used protocol for configuring  VLANs is IEEE 802.1Q.  VLANs are created by adding 32 bytes of data (a “tag”) to the header portion of the  ethernet frame. This allows the device to identify which VLAN a particular frame is  associated with. VLANs are identified numerically from 1 to 4096 (the last 12 bits of  the VLAN tag).    Trunk links are required to pass VLAN information between switches. A port on a Cisco  switch is either an access port or a trunk port.  Access ports – This switch ports belongs to carry the traffic of only one vlan. By default, it will carry the  traffic of native vlan (VLAN 1)  Trunk ports –  Switch port that can carry more than one VLAN traffic from one switch to another switch |
| **VLAN TAG** | When multiple VLANs travel in a trunk line they are tagged with thier respective  VLAN tags so that the receiving devices know which packet belongs to which VLAN.  The VLAN tag was invented to distinguish among different VLAN broadcast domains  on a group of LAN switches. The VLAN tag is a two-byte field inserted between the  source MAC address and the Ethertype (or length) field in an Ethernet frame.  Another two-byte field, the Tag Protocol Identifier (TPI or TPID), precedes the  VLAN tag field.  Used 802.1ad Ethernet Frame 802.1ad with double VLAN tagging is... |  Download Scientific Diagram  Native VLAN – IT Tips for Systems and Network Administrators |
| **OSPF Support for Traffic Engineering** | Traffic engineering allows you to control the path that data packets follow, bypassing  the standard routing model, which uses routing tables. Traffic engineering moves  flows from congested links to alternate links that would not be selected by the  automatically computed destination-based shortest path.  OSPF uses Type 10 LSAs to collect TE information in an area, such as the bandwidth,  priority, and link metrics. After processing the collected TE information, OSPF  provides it for CSPF to calculate routes.  When you enable traffic engineering for OSPF, the shortest-path-first (SPF) algorithm  takes into account the various label-switched paths (LSPs) configured under MPLS  and configures OSPF to generate **opaque link-state advertisements (LSAs)** that  carry traffic engineering parameters. The parameters are used to populate the traffic  engineering database. The traffic engineering database is used exclusively for  calculating explicit paths for the placement of LSPs across the physical topology.  The Constrained Shortest Path First (CSPF) algorithm uses the traffic engineering  database to compute the paths that MPLS LSPs take. RSVP uses this path  information to set up LSPs and to reserve bandwidth for them. |
| **Dijkstra’s algorithm** | Dijkstra’s algorithm is the better algorithm for the acyclic graph in which there is no  negative edge. Dijkstra’s have wasting a lot of memory and it cannot handle the  **negative edges**. |
| **sysroot** | The "sysroot" is the location the cross compiler will look for header files and libraries.  The sysroot directory acts as if it is the root of the system, |
| **EMake** | Electric Make® (”eMake”), is a new Make version .You can invoke  eMake interactively or through build scripts. Electric Accelerator is a software build  accelerator that dramatically reduces software build times by distributing the  build over a large cluster of inexpensive servers. |
| **File descriptor** | 1. One basic concept of Linux (actually Unix) is the rule that everything in 2. Unix/Linux is a file. Each process has a table of file descriptors that point to 3. files, sockets, devices and other operating system objects.      1. Stdin FD is 0 , Stdout FD is 1 ,stderr FD is 2 |
| **Select/poll/epoll** | simple solution is to create a thread (or process) for each client , block on read  until a request is sent and write a response. This is working ok with a  small number of clients but if we want to scale it to hundreds of clients, creating a  thread for each client is a bad idea IO Multiplexing The solution is to use a kernel mechanism for polling over a set of file descriptors.  There are 3 options you can use in Linux: select, poll,epoll   1. socket is a special file used for inter-process communication 2. I/O multiplexing  allowing to examine and block on multiple I/O streams 3. File descriptor is a integer used to index into a table of files that a process has   open.  **Select (), poll (), epoll () three sets of I / O multiplexing system calls can listen**  **to multiple file descriptors at the same time. They will wait for the timeout**  **specified by the timeout parameter until an event occurs on one or more**  **file descriptors.**  All the above methods serve the same idea, create a set of file descriptors , tell the  kernel what would you like to do with each file descriptor (read, write, ..) and use  one thread to block on one function call until at least one file descriptor requested  operation available  **1)select () system call**  **select does not bind the file descriptor and event. The select system call**  **monitors three sets of file descriptors**   1. readfds, 2. writefds and 3. exceptfds.   **1)poll () system call**  **Poll is somewhat smarter. It bind both file descriptors and events Using pollfd**  **structure**  **3, epoll () system call**  epoll is a [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel) [system call](https://en.wikipedia.org/wiki/System_call) for a scalable I/O to achieve **better performance** in more  demanding applications, where the **number of watched**[**file descriptors**](https://en.wikipedia.org/wiki/File_descriptor)**is large** (unlike the  older system calls, which operate in [O](https://en.wikipedia.org/wiki/Big_O_notation)(n) time, epoll operates in O(1)   1. epoll instance are in kernel |
| **Asymptotic notation** | Asymptotic notation, in computer science, big O notation is used to classify algorithms  according to how their running time or space require grow as the input size grows. |
| **Binary logarithm** | Problems  If there are 2 loops, Outer loops is incrementing sequentially , and inner loop is  incrementing by multiplication of 2 , then complexity of double for loop is n log n.  O(1) is better than linear time O(n) because the former is not depending on the  input-size of the problem. |
| **BST** | the height of a binary tree is equal to the largest number of the edges from the root to the most  distant leaf node.  A similar concept in a binary tree is the depth of the tree. The depth of a node in a binary tree  is the total number of edges from the root node to the target node.  The diameter of a binary tree is the length of the longest path between any two  nodes in a tree. This path may or may not pass through the root. The length of path  between two nodes is represented by the number of edges between them. |
| **NVD** | National Vulnerability Database (NVD) is a government repository of  standards-based vulnerability information. |
| **BST/AVL**  **Complexity** | search ,insertion and deletion in  BST have worst case time complexity of O(n).  However, AVL tree has worst case time complexity  of O(logn). |
| **private IP address** | Port forwarding allows remote computers (for example, computers on the Internet)  to connect to a specific computer or service within a private local-area network  (LAN).  Rogers:  <http://192.168.0.1/login.html>  Username: **cusadmin,**Password:**password**(or your current Wi-Fi password) |
| **Linux priority top command** | Linux Kernel implements two separate priority ranges –   * The nice value range is -20 to +19 where -20 is highest, 0 is default/   Nice value: minus 20 to plus 19; larger (+19) nice correspond to lower priority.   * Real-time priority: 0 to 99; higher real-time priority values correspond to   a greater priority. PR = 20 plus nice ,  0 is default nice priority of process  The sched\_setscheduler() system call set scheduling policy of thread to real time  SCHED\_FIFO  SCHED\_RR |

|  |  |
| --- | --- |
|  | 滴水穿石- OSDI - Viva La Vida |
| **MacSec** | Media Access Control security (MACsec) provides point-to-point security on  Ethernet links. MACsec is defined by IEEE standard 802.1AE. You can use  MACsec in combination with other security protocols, such as IP Security (IPsec)  and Secure Sockets Layer (SSL), to provide end-to-end network security.  First of all, MACsec and IPsec operate on different network layers. IPsec works on  IP packets, at layer 3, while MACsec operates at layer 2, on ethernet frames. Thus,  MACsec can protect all DHCP and ARP traffic, which IPsec cannot secure. On the  other hand, IPsec can work across routers, while MACsec is limited to a LAN.  With both MACsec and IPsec, user applications do not need to be modified to take  advantage of the security guarantees that these standards provide. |
| **DPDK** | DPDK (Data Plane Development Kit) is a framework (under the Linux Foundation)  comprised of various userspace libraries and drivers for fast packet processing.  Originally developed by Intel to run on x86 based CPUs, DPDK now supports other  CPU types. DPDK leverages existing Intel Processor technologies like SIMD instructions  (Singles Instruction Multiple Data), Huge-pages memory, multiple Memory channels  and Caching to provide acceleration with its own libraries.  Though DPDK uses a number of techniques to optimise packet throughput, how it  works (and the key to its performance) is based upon Fast-Path and PMD.  **Fast-Path (Kernel bypass)** - A fast-path is created from the NIC to the  application within user space, in turn, bypassing the kernel. This eliminates  context switching when moving the frame between user space/kernel space.  Additionally, further gains are also obtained by negating the kernel  stack/network driver, and the performance penalties they introduce.  **Poll Mode Driver** - Instead of the NIC raising an interrupt to the CPU when a frame is  received, the CPU runs a poll mode driver (PMD) to constantly poll the NIC for new  packets. However, this does mean that a CPU core must be dedicated and assigned to  running PMD. |
| How https works | 1>When you enter the URL www.Google.com, Google’s server gives its public key and  Digital certificate (which was signed by GeoTrust) to the Browser.  2) Now browser has to verify the authenticity of the certificate i.e. it’s actually signed  From GeoTrust or not. As browsers come with a pre-installed list of public keys from  all the major CA’s, it picks the public key of the GeoTrust and tries to decrypt the digital  signature of the certificate which was encrypted by the private key of GeoTrust.  3) If it’s able to decrypt the signature (which means it’s a trustworthy website) then it  proceeds to the next step else it stops and shows a red cross before the URL.  As I mentioned, Google sends its public key when you enter www.Google.com . Any  data encrypted with this public key can only be decrypted by Google’s private key which  Google doesn’t share with anyone.  2) After validating the certificate, browser creates a new key let’s call it Session Key  ( symmetric key) and make 2 copies of it. These keys can encrypt as well as decrypt the  data.  3) The browser then encrypts (1 copy of session key + other request data) with the  Google's public key . Then it sends it back to the Google server.  4) Google’s server decrypts the encrypted data using its private key and gets the session  key , and other request data. Now, see, server and browser both have got the same copies  of session key of the browser. No one else has this key, therefore, only server and  browser can encrypt and decrypt the data. This key will now be used for both to decrypt  and to encrypt the data.  5) If the user closes the website and opens again, a new session key would be created. |
| Open stack | Open stack is an opensource cloud platform alternative to proprietary solutions like  Amazon AWS and Microsoft azure . Its an operating System for a cloud infrastructure.  It is a management layer on top of all other components like hypervisors, SDN, VM  images, Storage, etc.  It is a software which help in creating private or public clouds. It, consists of a  many inter-related projects for networking, storage, computing which can be deployed  over the set of systems and a cloud infrastructure can be created, then this infrastructure  can be used or rented to others for usage similar to how Amazon Clouds Services are. It  can easily run-on commodity hardware and scale horizontally (just add more commodity  hardware or PC) |
| RTOS | Properties of RTOS   * time critical application * **preemptive scheduling policy,** * simplify the OS, many general-purpose functions (disk I/O) omitted |
| Moore's Law | Moore’s Law is a computing term which originated around 1970; the simplified version  of this law states that processor speeds, or overall processing power for computers will  double every two years. To break down the law even further, it specifically stated that  the number of transistors on an affordable CPU would double every two years.  Network capacity. Butter's law says that the amount of data coming out of an optical fiber  is doubling every nine months. Thus, the cost of transmitting a bit over an optical  network decreases by half every nine months.    Optical networking and dense wavelength-division multiplexing (DWDM) is rapidly  bringing down the cost of networking, and further progress seems assured . |
| Data center fabric | A data center fabric is a system of switches and servers and the interconnections  between them that can be represented as a fabric. .Cisco offering includes fabric  management via Application Policy Infrastructure Controller (APIC) or  Data Center Network Manager (DCNM) |
| C plus Plus | A class is an expanded concept of a data structure: instead of holding only data, it  can hold both data and functions.  ▪An object is a class variable or an "instance" of a class .  ▪OOP is a design philosophy where program is divided into parts called objects.  ▪Object Oriented Programming(OOP) employs the **bottom-up programming** approach.  ▪In Procedure oriented programming (POP), program is divided into small functions  ▪POP follows Top-Down approach  The only difference between a struct and class in C++ is the default accessibility  of member variables and methods. In a struct they are public; in a class they are private.  **Polymorphism** : more than one function with same name and different working,  In static polymorphism(**function overloading , templates ,default arguments and**  **operator Overloading**) memory will be allocated at compile time. In run time  polymorphism (Virtual function )memory will be allocated at runtime.  **Data** **Abstraction** : hide inner details by making class members private  the quality of dealing with ideas rather than events.  **Encapsulation** : wrapping data members and member functions in a single unit called  class. Data is only accessible through the object of the class.    **Inheritance** : access the properties and features of base class into derived class.  1>A derived class with only one base class is called **single inheritance.**  2>A derived class with multiple base class is called **multiple inheritance.**  3>A derived class with two base classes and these two base classes have one common  base class is called **multipath inheritance.**  **virtual base class**  In multipath inheritance. In CHILD class have two copies of Base class. This problem is  also called as DIAMOND Problem.  This can be solved by declaring the common base class as a virtual base class  Now only one copy of the members of grandparent will be inherited into child  **C++ storage classes?**  **Automatic variable** ,stack segment, initial value garbage  **External variable** ,data segment, initial value 0  **static variable** ,data segment, initial value 0  **register variable** ,data segment, initial value garbage  **What are storage qualifiers in C++ ?**  Const ,volatile  **iostream**  is used to access the input output built in functions of the language same as stdio in c  **Using Namespace std**  C++ has a standard library that contains common functionality for building your  applications like such as cin,cout,string or vector, algorithms. Import the entirety of the  std namespace into the current namespace of program to avoid duplicates function  without having to prefix std:: before each of these functions.  **Structure/class**  Structure default access type is public , but default class access type is private.  A structure is used for grouping data whereas class can be used for grouping data and  methods.  **Protected and private members**  Protected access modifier is similar private access modifiers, the difference is that  but they can be accessed by any subclass(derived class) of that class.  **Friend function**  - A friend function of a class is defined outside that class' scope but it has the right  to access all private and protected members of the class ...  **Constructor**  A constructor is a member function with the same name as its class. A constructor is  different from normal functions in following ways: Constructors don’t have return type.  Constructor creates an object and initializes it. **It also creates vtable for virtual**  **functions.**  **Default Constructor**  Constructor with no arguments or all the arguments has default values.  **copy constructor**  A copy constructor is a member function which initializes an object using  another object of the same class.  Point p1(10); // Normal constructor is called here . Shallow copy  Point p2(p1); // Copy constructor is called here . Shallow copy  Point p2 = p1; // Copy constructor is called here . Shallow copy  **Copy constructor and assignment operator**, are the two ways to initialize one object  using another object. The fundamental difference between the copy constructor and  assignment operator is that the**copy constructor** allocates separate memory to both  the objects, i.e. the newly created target object and the source object.  The **assignment operator** allocates the same memory location to the newly  created target object as well as the source object.   1. Default copy constructor does only shallow copy. 2. Deep copy is possible only with user defined copy constructor**.**   Copy constructors are called in following cases:  (a) when a function returns an object of that class by value  (b) when the object of that class is passed by value as an argument to a function  (c) when you construct an object based on another object of the same class  (d) When compiler generates a temporary object  **Destructor** is a member function which destructs or deletes an object. A destructor  function is called automatically when the object goes out of scope: the function ends, the  program ends, a block containing local variables ends, a delete operator is called  Destructors have same name as the class preceded by a tilde (~),Destructors don’t take  any argument and don’t return anything  a destructor cannot be overloaded, and it has the only form without the parameters.  **Virtual function**  A virtual function is a member function which is declared with a virtual keyword  In a base class and is re-defined by a derived class. When you point to a derived class  object using a pointer to the base class, you can call a virtual function for that  object and execute the derived class’s version  base\* bptr;  derived d;  bptr = &d;  **Abstract class**  sometimes implementation of all function cannot be provided in a base class because  we don’t know the implementation. Such base class is called abstract base class . An  abstract class is a  class in C++ which have at least one pure virtual function. A pure  virtual function is declared by assigning 0 in declaration.    **vtable**  Virtual functions are implemented using a table of function pointers, called the  vtable. There is one entry in the table per virtual function in the class. This table is  created by the constructor of the class. When a derived class is constructed, its base  class is constructed which creates the vtable. If the derived class overrides any of the  base classes virtual functions, those entries in the vtable are overwritten by the  derived class constructor. This is why you should never call virtual functions from a  constructor: because the vtable entries for the object may not have been set up by  the derived class constructor yet, so you might end up calling base class  implementations of those virtual functions  Why there are no virtual constructors in C++?  Constructor cannot be virtual because when constructor of a class is  executed there is no vtable in the memory means no virtual pointer defined yet.  Virtual destructor :you destroy an object through a pointer to a base class, and the  base-class destructor is not virtual, the derived-class destructors are  not executed, and the destruction might not be complete.  So the base class destructor should be virtual to execute the destructors from derived  to base class order.The **insertion** (<<) **operator**,  **Static data member**  There is only one copy of the static data member in the class,  even if there are many class objects. ... The static data member is always  initialized to zero when the first-class object is created.  **Static member function**  By declaring a function member as static, you make it independent of any  particular object of the class. A static member function can be called even  if no objects of the class exist and the static functions are accessed using  only the class name and the scope resolution operator ::.  **Question: Can we have a recursive inline function in C++?**  **Answer**:Even though it is possible to call an inline function from within itself in C++,  the compiler may not generate the inline code. This is so because the compiler won’t be  able to determine the depth of the recursion at the compile time.  **Question: Explain ‘this’ pointer?**  **Answer**: The ‘this’ pointer is a constant pointer and it holds the memory address of the  current object. It passes as a hidden argument to all the nonstatic member function calls.  Also, it is available as a local variable within the body of all the nonstatic functions.  As static member functions can be called even without any object, i.e. with the  class name, the ‘this’ pointer is not available for them  class ClassName {  private:  int dataMember;  public:  method(int a) {  // this pointer stores the address of object obj and access dataMember  this->dataMember = a;  ... .. ...  }  }  int main() {  ClassName obj;  obj.method(5);  }  **Operator overloading**  which an operator is overloaded to give user defined meaning to it. ...  For example '+' operator can be overloaded to perform addition on  various data types, like for Integer, String(concatenation) etc. |
| Exception handling in C++ | Exception handling in C++ consist of three keywords: try, throw and catch:    char\* is a mutable pointer to a mutable character/string.const char\* is a mutable pointer  char\* is a **mutable** pointer to a **mutable** character/string.  const char\* is a **mutable** pointer to an **immutable** character/string. You cannot change the  contents of the location(s) this pointer points to |
| Copy constructor  And operator overloading | t1 = t2 operator over loading |
| New /malloc | 1. For a dynamically allocated object, constructor is invoked by new operator   Example geeks obj = new geeks()  geeks obj = new geeks[4]; 4 objects created , to free used  delete [] geeks   1. New doesn’t need type casting, Malloc requires typecasting 2. New is a operator it can be overloaded, Malloc is a function cannot be overloaded 3. New does not require to explicitly specifying the quantity of memory allocated, 4. If the new operator fails to allocate the memory, it throws an exception   that must be handled by the code else the program will terminate.   1. new is related to delete , malloc is related to free   Once the memory is allocated using a new operator, it can’t be resized |
| template | Templates are expanded at compiler time. This is like macros. The difference is, compiler Does type check before template expansion. The simple idea is to passdata type as a parameter so that we don’t need to write the same code for different datatypes.For example, a software company may need sort() for different data types.   **Function Templates** We write a generic function that can be used for different data types.  Examples of function templates are sort(), max(), min(), printArray().  **Class Templates** Like function templates, class templates are useful when a class defines  something that is independent of the data type. Can be useful for classes like  LinkedList, BinaryTree, Stack, Queue, Array, etc. |
| Mutable keyword | Mutable data members are those members whose values can be changed in  runtime even if the object is of constant type. It is just opposite to constant. |
|  | The Standard Template Library (STL) is a set of C++ template classes to provide  common programming data structures and functions such as lists, stacks, arrays, etc.    container classes store objects and data and it has methods for accessing its  elements. In particular, every type that is a model of Container has an associated iterator  type .iterators is like a pointer and provide a means for accessing data stored in  container classes  such a vector, map, list, etc  **Sequence Containers**  [Sequence containers](http://en.cppreference.com/w/cpp/named_req/SequenceContainer) are used for data structures that store objects of the same type in  a linear manner. The STL Sequence Container types are:  The STL SequenceContainer types are:   * [array](http://en.cppreference.com/w/cpp/container/array) represents a static contiguous array * [vector](http://en.cppreference.com/w/cpp/container/vector) represents a dynamic contiguous array * [forward\_list](http://en.cppreference.com/w/cpp/container/forward_list) represents a singly-linked list * [list](http://en.cppreference.com/w/cpp/container/list) represents a doubly-linked list * [deque](http://en.cppreference.com/w/cpp/container/deque) represents a double-ended queue, where elements can be * added to the front or back of the queue.   While [std::string](https://embeddedartistry.com/blog/2017/7/24/stdstring-vs-c-strings) is not included in most container lists, it does in fact meet  the requirements of a SequenceContainer.  **Adapters Containers**  They are wrappers around other container types (such as a vector, deque, or list).  Example Stack (LIFO), queue(FIFO), [priority queue](http://en.cppreference.com/w/cpp/container/priority_queue) .a priority queue is an abstract  data type similar to regular queue or stack data structure in which each element  additionally has a "priority" associated with it. In a priority queue, an element  with high priority is served before an element with low priority largest element  (by default)    **Associative Containers**  The C++ library provides eight associative containers Associative containers store  elements which are retrieved by a key  **Ordered Set and Map** are balanced binary search tree ([Red-Black Tree](https://www.quora.com/topic/Red-Black-Tree)) Thus, you use them  when you want insertion ,searching, deletion in O(log n). They don't allow duplicates.  the elements are kept in order of the keys (ascending by default), which  sometimes can be useful.  Example set<int> s; store counters 1 to 100  Student marks and roll number, The roll number is the key and  each student has a different roll number, hence unique key representing them.  Example map<int, int> marks;  A common problem for hash tables is that the worst case cannot be guaranteed  (though the likelihood is very small), which renders it extremely vulnerable in  cyber-attacks. That’s why hash maps are not appreciated in systems demanding  real-time performance.  Unordered Set and Map are an implementation of [Hash Tables](https://www.quora.com/topic/Hash-Tables). So, you use them  when on average you want insertion, search, deletion in O(1). Though, worst case will  O(n).  Unordered map is more efficient that map, as it’s based on hash tables, which  provides insertion and deletion time complexity close to O(1). However, it does  not sort the keys as map does. |
| **Q&A** | A dangling pointer arises when you use the address of an object after its lifetime is over.    **References are less powerful than pointers**  1) Once a reference is created, it cannot be later made to reference another object;  2) References cannot be NULL.  3) A reference must be initialized when declared.  What do you mean by Stack unwinding?  It is a process during exception handling when the destructor is called for all local objects  between the place where the exception was thrown and where it is caught.  If B is a **shallow copy** of A . *B and A point to the same memory location.* If B is a  **deep copy** of A, *B and A point to different memory locations*  What is inline function?  The inline, substitution occurs only at the compiler’s discretion. Can inline functions cannot  have recursion?  Overriding a method means that replacing a method functionality in derived class.  Explain the scope resolution operator?  It permits a program to reference an identifier in the global scope that has been  hidden by another identifier with the same name in the local scope.  Manipulators are the functions which can be used in conjunction with the insertion  (<<) and extraction (>>) operators on an object. Examples are endl and setw. |
| **Smart pointer** | In C++ 11, it introduces smart pointers .A smart pointer is a container class that wraps  a 'raw' (or 'bare') C++ pointer, to manage the lifetime of the object being pointed to.  Smart pointers should be preferred over raw pointers. With raw pointers, the programmer has  to explicitly destroy the object when it is no longer useful. |
| **Lambda Function** | C++ 11 introduced lambda expression to allow us write an inline  function which can be used for short snippets of code that are not  going to be reuse and not worth naming. In its simplest form  lambda expression can be defined as follows: |
| **Move constructor** | |  | | --- | |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |
| **Design patterns** | Design patterns are documented tried and tested solutions for recurring problems in a  given context.  The singleton pattern is used to limit creation of a class to only one object. Examples  Including caches, thread pools, and registries.  A log file is the big one. You don't want to just abandon a single log file. You want to  flush, sync and close it properly. This is an example of a single shared resource  that has to be manage    You can create a singleton class by making its constructor as private, so that  you can restrict the creation of the object. Provide a static method to get instance  of the object, wherein you can handle the object creation inside the class only.  Major disadvantages 1. an inhibiting feature when more than one instance is required.  2. Difficult to make thread-safe. 3. Global access to singleton object is a bad design practice. 4. Singleton does not support inheritance of the class.  5 make unit testing very hard  **Factory Method**  Factory pattern says that object creation will be handled by subclass class  using the input type. We are building an application that needs to support two different  databases, Oracle and SQL Server.  Whenever we insert a data into a database, we create a SQL Server– specific connection  or an Oracle server– specific connection and then we can proceed. If we put these codes into  if-else (or switch) statements, we may need to repeat a lot of code.  This kind of code is not easily maintainable because whenever we need to support a new type of  connection, we need to reopen code and place the modifications. A factory method pattern  focuses on solving similar problems in application development. |
| Memory Layout of C Programs | <https://cdncontribute.geeksforgeeks.org/wp-content/uploads/memoryLayoutC.jpg>  **From low to high memory**  **1 Text Segment:** contains executable instructions.  **2. Data Segment:** contains global & static variables  3. heap goes up  4. stack goes down |
| Dynamic Programming | 1. In Dynamic Programming, a problem is divided into sub-problems, 2. Solutions of these sub-problems combined together overall solution. 3. Dynamic Programming solve each of these sub-problems just once. 4. Memorization is a term describing an optimization technique where you   cache previously computed results, and return the cached result when the same computation is needed again |

|  |  |
| --- | --- |
| Mutex V/s Semaphore | 1> In **mutexes has an ownership property**, only the **thread** that took the  **lock** has the key. Only that thread alone can release the lock. Binary semaphores  doesn’t have an ownership property, as any thread can take the key to open the  lock.  2)A semaphore is a generalized mutex. In lieu of single buffer, we can **split**  the 4 KB buffer into four 1 KB buffers (identical resources). A semaphore  can be associated with these four buffers. The consumer and producer  can work on different buffers at the same time. |
| virtualization | A software container is a way to bundle and isolate processes (software) running  on a server. Virtual machines and containers differ in several ways, but the primary  difference is that containers provide a way to virtualize an OS so that multiple  workloads can run on a single OS instance. With VMs, the hardware is being  virtualized to run multiple OS instances. |
| Docker/ Kubernetes | |  | | --- | | Docker is what enables us to run, create and manage containers on a single host  Kubernetes can then allow you to automate container provisioning, networking,  load-balancing, security and scaling across all these nodes from a single command line  or dashboard.  A collection of nodes that is managed by a single Kubernetes instance is referred to  as a Kubernetes cluster.  Now, why would you need to have multiple nodes in the first place? The two main  motivations behind it are:  1. To make the infrastructure more robust: Your application will be online, even if  2. some of the nodes go offline, i.e, High availability.  3. To make your application more scalable: If workload increases, simply spawn more  4. containers and/or add more nodes to your Kubernetes cluster.  Kubernetes automates the process of scaling, managing, updating and removing  containers. In other words, it is a container orchestration platform. While Docker is  at the heart of the containerization, it enables us to have containers in the first place.  Differences Between Kubernetes and Docker In principle, Kubernetes can work with  any containerization technology. | |
| ASLR | Address space layout randomization (ASLR) is a memory-protection process for  operating systems (OSes) that guards against buffer-overflow attacks by randomizing  the location where system executables are loaded into memory.  The success of many cyberattacks, particularly zero-day exploits, relies on the  hacker's ability to know or guess the position of processes and functions in memory.  ASLR is able to put address space targets in unpredictable locations. If an  attacker attempts to exploit an incorrect address space location, the target application  will crash, stopping the attack and alerting the system.  ASLR works alongside virtual memory management to randomize the locations of  different parts of the program in memory. Every time the program is run, components  (including the stack, heap, and libraries) are moved to a different address in virtual |
| Object Size Checking (OSC) | Object Size Checking (OSC) leverages a builtin compiler technique to determine  buffer overflows in C/C++ code. various optimization passes enabled with -O2 |
| xspace | Making the stack (and heap) non-executable provides a high degree of protection  against many types of buffer overflow attacks for existing programs.  is that execution occurs in the code section, which is neither stack nor heap. |
| GPB | Google has a tremendous amount of internal traffic. It makes sense for  the company to optimize communication between internal services to make it as efficient as  possible, as this directly translates to monetary savings (fewer CPU cycles spent on  communication = less energy consumed) as well as more responsive public-facing  applications and services.  The serialization problem has been solved in other ways, of course. XML is a very  robust data serialization format which provides most of the benefits you would want.  It is standardized, and libraries have naturally been built in every major programming  language to parse and generate it. JSON is similarly popular, though not quite as  expressive (but even more concise and, arguably, more readable).  But XML and JSON have some baggage. For one, both are designed to be human  readable. This of course requires that they be text-based, which means that for the  purpose of transporting messages in either format data must be encoded on one end and  then decoded on the other end. They are also self-describing. This is a big benefit, but  it also necessarily increases message size, since some sort of schema information needs  to be included in the message itself.  Protocol buffers are not self-describing. Both parties (sender and receiver) must have a  shared schema in order to understand what a message means. This is fine for internal  service calls (it's actually also fine for a public API, though it requires more care), and it  allows messages to be much more compact—which of course contributes to throughput  (obviously, it takes less time to transmit a smaller message than a larger one). They are  also binary, not text-based, which improves efficiency by eliminating the encoding and  decoding steps.  So protocol buffers provide a lot of value as a data serialization format, in the same  way that other serialization formats like XML and JSON do. But they also offer much  greater efficiency, which is a significant part of the reason Google developed and uses them.  [Protocol buffers](https://developers.google.com/protocol-buffers/) are a language-neutral way of serializing structured data. If two applications want to communicate structures (with multiple fields) over  a wire / network, the structure is serialized using Protocol Buffers   * They are a simple interface for serializing structured data. * They are 3 to 10 times smaller than XML. * They are 20 to 100 times faster than XML to parse and serialize. * The protoc compiler generates easy-to-use data classes in many client languages.   As you can imagine, |
| Reader-writer lock | When a writer is writing the data, all other writers or readers will be blocked  until the writer is finished writing. Readers–writer locks are usually constructed  on top of [mutexes](https://en.wikipedia.org/wiki/Mutex) and  [condition variables](https://en.wikipedia.org/wiki/Condition_variable), or on top of [semaphores](https://en.wikipedia.org/wiki/Semaphore_(programming)). |
| **STACK** | **Key Differences Between Stack and Heap Allocations**   1. In a stack, the allocation and deallocation is automatically done by   whereas, in heap, it needs to be done by the programmer manually.   1. Memory shortage problem is more likely to happen in stack whereas   the main issue in heap memory is fragmentation.   1. Stack is not flexible, the memory size allotted cannot be change |
| **mutual exclusion** | A **mutual exclusion** (mutex) is a program object that prevents simultaneous  access to a shared resource (critical section) by two or more threads |
| **Deadlock** | **Deadlock can arise if following four conditions hold simultaneously**  ***Mutual Exclusion:*** One or more than one resource are non-sharable  (Only one process can use at a time) ***Hold and Wait:*** A process is holding at least one resource and waiting for resources. ***No Preemption:*** A resource cannot be taken from a process unless the process releases  the resource. ***Circular Wait:*** A set of processes are waiting for each other in circular form. |
| How switch works | A switch kicks off with an empty table that maps MAC address to the outgoing  switch port. When a switch receives a packet at the first time, it adds the  source MAC along with the port at which it received the packet to the switch  table. As it doesn’t have an entry for the destination MAC yet, it broadcasts  that packet to all the devices connected in local network. Once the destined device  responds, switch receives this packet and update the switch table.  This is how switch updates its table and uses it to route packets to  corresponding switch port. |
| ECMP algorithms | There are probably a dozen different ECMP algorithms.  They also vary based on  whether you are using Layer 2 or Layer 3. Fundamentally, <n> paths exist in the forward and reverse path direction.  A modulo <n> hash of various header information is performed, and the output |
| Backup path | IGP pre-computes a backup path per primary path per IGP destination: per-path.  The backup path is pre-installed in data plane. Upon local failure, all the  backup paths of the impacted destinations are enabled in a  prefix-independent manner (<50msec loss of connectivity)  **pure backup path** Backup path which is not ecmp  **icmp backup path** Backup path which is ecmp  **protected path** is a active path which has backup path  **LFA**  When backup path are directly connected  **RLFA(P,Q node**) When backup path are not directly connected ,packet will  be encapsulated till it reaches p,q router  **TILfA**  When backup path are not directly connected when  P and q are disjoint |
| OS concept | Page Fault: A page is a fixed-length block of memory that is used as a unit of  transfer be­tween physical memory and external storage like a disk, and a  page fault is an interrupt (or exception) to the software raised by the hardware  when a program accesses a page that is mapped in address space, but not  loaded in physical memory. |
| DMA | Direct Memory is a feature which provides direct access (read/write) to system  memory with­out interaction from the CPU. using “DMA Controller” |
| STACK protection | in a multi-threaded environment, there can be multiple stacks in a process.  One threat to the stack is malicious program input, which can overflow a buffer  and over­write stack pointers, simple method GCC, you use -fstack-protector-all. |
| Routing protocols | BGP is the core routing protocol of the Internet.“When a BGP router first comes  up on the Internet, it does is down­load the entire routing table of each  neighboring router. After that it only exchanges much shorter update messages |
| Virtualization | With VMs, the hardware is being virtualized to run multiple OS instances.  Docker is an open source project that makes it easier to create, deploy and run  applications.in containers The applications are packaged in a docker container  which contains all the dependencies (libraries, packages) that are needed to  deploy the application. By using docker, an application can be easily  moved around from the developer’s laptop, into the testing  environment and finally into production.  .  **Kubernetes pods** :A Kubernetes pod is a group of containers that are  deployed together on the same host. If you frequently deploy single container  s, you can generally replace the word "pod" with "container" and accurately  understand the concept.  **Docker** is a run time engine running on your computer. It’s a daemon that is  in charge of containers start, stop on that single computer. So Docker is  about managing works within a single machine.  **Kubernetes is kind of a cluster management software**. It is a group of  daemons that is in charge of a cluster of machines. Though there is a single  daemon (kubelet) running on an individual machine, the kubelet by itself  does not have much value on the table; it is these group of kubelets (  along with kubernetes controllers that control them) make decisions about  the whole cluster. So k8s is about managing works for a cluster of machines. |
| Load balancing | **layer 3** load-balancer takes routing decisions based on IP alone (source & dest)  **layer 4** load-balancer takes routing decision based on IPs **and** TCP or UDP ports.  **Layer 7**load balancer operates at the high-level *application* layer, which deals with  the actual content of each message.Layer 7 load balancing is more CPU-intensive rarely causes degraded performance on a modern server |
| Machine learning | Machine learning is an application of artificial intelligence (AI) that provides  systems the ability to automatically learn and improve from experience  without being explicitly programmed.  In **Supervised** learning, you train the machine using data which is  well "labeled.  **Unsupervised** learning is where you only have input data and no  corresponding output variables.  - |
| Intent-based networking | Intent-based networking, or IBN, uses advanced algorithms, AI-ML and network  orchestration to make your enterprise network smarter. But with IBN, an intent  can automatically be interpreted and applied throughout the network using GUI |
| Incognito mode | Incognito mode (Chrome), InPrivate (IE & Edge), Private Window (Firefox)  Assume that you are going to an internet cafe to look up your mail.  Incognito mode doesn't save passwords, cookies and your browsing history. So  it is safe to use incognito mode if you are using some stranger's system. |
| Broadcom | Broadcom is the market leader for switching merchant silicon’s used in  network switches. There are other vendors such as Mellanox, Marvel,  EZChip, Microsemi and Cavium,  **1- Strata XGS Family includes the following silicons**  Trident 2 (10g / 40G)  Trident 2+ (10G / 40G , Enhanced Trident 2 , with VXLAN Routing)  Tomahawk (100G)  **2- Strata DNX Family (Also known as Dune family)**  Qumran (10G/40G)  Jericho (packet processor)  FE 3600 (Fabric)  This family is pretty new and is based on Broadcom acquisition of Dune  networks. Feature wise, DNX comes from a different architecture has some  benefits and advantages :   1. Expandable TCAM : 2. Expandable Packet Buffer : 3. Jericho+, Jericho2 delivers 5X higher bandwidth at 70% lower |
| Libvert | libvirt is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [API](https://en.wikipedia.org/wiki/Application_programming_interface), [daemon](https://en.wikipedia.org/wiki/Daemon_(computing)) and management tool for managing  [platform virtualization](https://en.wikipedia.org/wiki/Platform_virtualization). It can be used to manage [KVM](https://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine), [Xen](https://en.wikipedia.org/wiki/Xen), [VMware ESXi](https://en.wikipedia.org/wiki/VMware_ESXi),  [QEMU](https://en.wikipedia.org/wiki/QEMU) and other virtualization technologies. These APIs are widely used in the  Orchestration layer of [hypervisors](https://en.wikipedia.org/wiki/Hypervisor) in the development of a cloud-based solution. |
| macro | 1. Macros are expanded by the preprocessor before compilation takes place.  2. The compiler decides which functions to inline.  3. The advantage of using macro is the execution speed  4. The disadvantage of the macro is the size of the program. |
| Anycast | **Unicast** is one-to-one.  **Broadcast** is one-to-many. It only works on a local network.  **Multicast** is more complicated. It's one-to-many, but it can go outside a  local network - even across the internet  **Anycast** is a special case of unicast, as if you wrote a take-out order to  "McDonalds, New York" and the letter carrier just delivered it to the first one  he saw because they all had the same address. It's one-to-one, you just  don't know or care which one |
| Penultimate hop popping (PHP) | PHP is penultimate hop popping which means remove the label one hop before  its destination. It refers to the process whereby the outermost label of an MPLS  tagged packet is removed by a Label Switch Router (LSR) before the packet is  passed to an adjacent Label Edge Router. The process is important in a Layer 3  MPLS VPN environment as it reduces the load on the LER. If this process didn't  happen, the LER would have to perform at least 2 label lookups: Both implicit and explicit null labels are generated by last hop router to itsneighbors.Implicit null is by default which means penultimate router shouldonly send IP packet thus it pops the label The one disadvantage inimplicit null approach is if the network is configured for QoS based on MPLSEXP bits, then QoS is lost between penultimate router and last hop router.In this case, we can make use of Explicit null which means penultimate hoprouter does not pop the label. It sends with label value of 0 but with other fieldsincluding EXP bits intact.This way QoS treatment is preserved between penultimate router and lasthop router. Explicit null should be configured manually in last hop router. |
| BFD | BFD is a network protocol that is used to detect failures in the link between two  connected devices. BFD enables a session between the two ends of link, by a  simple authentication  Packets are transmitted at regular intervals. If the packets do not turn up, it  indicates that the link has failed. The time intervals can be configured  Although reducing the EIGRP, IS-IS, and OSPF timers can result in  minimum detection timer of one to two seconds, BFD can provide failure  detection in less than one second. |
| Pseudowire | Pseudowire (PW) is an emulation of a Point to Point wired connection (E1/T1)  overMultiprotocol Label Switching (MPLS) cores. |
| VXLAN | VXLAN is essentially a **tunneling technology.** VXLAN adds the VXLAN header to  an original data frame, encapsulates the frame into a UDP packet, and  forwards the UDP packet in traditional IP network transmission mode.  After the UDP packet arrives at the end point, the end point removes the outer  header and sends the original data frame to the target terminal. |
| NOP | NOP can be used in early board development. Need to know clock speed |
| Relational Database | A transaction can be defined as a group of tasks. A single task is the minimum  processing unit which cannot be divided further. Let's take an example of a  simple transaction. Suppose a bank employee  transfers Rs 500 from A's account to B's account.   1. taskTake money from A count 2. taktransfer to B account 3. task to pay transaction fees   Relational databases like Microsoft's SQL Server use locks ( DELETE, INSERT  and UPDATE )to prevent multiple users from making conflicting modifications  to a set of data: when a set of data is lockedby a user, no other users can modify  that same set of data until the first user finishes modifying the data and  relinquishes the lock. |
| Puzzles | Q1)There are three boxes, one contains only apples, one contains only oranges,  and one contains both apples and oranges. The boxes have been incorrectly  labeled such that no label identifies the actual contents of its box.  Opening just one box, and without looking in the box, you take out one piece  of fruit. By looking at the fruit, how can you immediately label all of the  boxes correctly?  Q2)You have a 3-gallon jug and 5-gallon jug measure exactly 4 gallons?  Q3)If you look at the clock and the time is 3:15, what is  the angle between the hours and the minutes hands?  The minute hand travels 6° per minute and the hour hand travels 0.5° per minute    At 3:15, the minute hand would have effectively travelled for 15 minutes  covering (15 \* 6 = 90°) after completing three full revolutions in three hours  with net ZERO displacement from reference point.  The Hour hand would have travelled for 180 + 15 = 195 minutes. It covers an angle  of 195 x 0.5 = 97.5°  The angle between the two hands now is 97.5°-90°=7.5°  12 persons 66 handshakes Let the number of persons= n  Number of handshakes by first person= n minus 1,  (As he cannot handshake with himself)  Number of handshakes by second person = n minus 2  (As he has already handshake with first person)  N(n-1)/2 =66 |
| **Managed timer** | Managed timer is a component in IOX that is responsible for maintaining timers  and firing them within the appropriate precision limits  issues:  1 Insertions into the linked list are expensive O(n) because the list is maintained sorted  and the list can potentially grow upto 50,000 nodes.  2 This is currently very CPU intensive and hence degrades the overall system performance.  3 how to handle jitter  4 timers that can be grouped and fired together are not done so  5 Low insertion and deletion time – O(1) because of the sorted nature.  Managed timer is a library that is responsible for maintaining timers and firing them within the  appropriate precision. At a given moment only 1 timer (root node) will be programmed in  kernel using timer\_settime API. example the parent has the lowest value of the timer of its leaf  nodes (1 min, 2 min, 3 min) so parent and root both will be 1 minutes   1. Talk about performance comparisons of using one data structure over another. 2. Design for backwards compatibility 3. Having to support a poorly designed and written legacy system. 4. redundant code in an outdated and obscure language with an awful design.   Both single and multi-level hash tables were considered. A lot of reordering was required with  every insert and delete operation. The data ranges could not be determined dynamically and so  only fixed ranges can be supported. Since not all timers follow the same range set as  well, i.e. timers right from milliseconds to a few hours exist and are not evenly distributed;  Binary Trees: The problem with complete binary trees is the amount of reordering required in  order to maintain the balance of the tree. This can take O(log n) operations for both inserts and  deletes which is not desired. |
| Jerico-2 | 1. Up to 10Tb/s switching capacity per device. 400G/100G Ethernet port density 2. Cost parity for comparable port Density 3. programmable **pipeline** 4. **Deep buffers**: Deep buffers (for burst handling), eliminating packet   Drops They allow the network to accommodate some level of  variation in the network traffic, simply put, a buffer is a reserved portion  of memory where “overflow” data can be stored levels allow |
| SSL | SSL (Secure Sockets Layer) is protocol to encrypt the information between client and server.  TLS (Transport Layer Security) is an improved version and the successor to the SSL protocol  which is also used to encrypt information between client and server. The two terms are used to  describe two different versions but SSL is more widely used and term SSL/TLS also conveys the  same purpose.HTTPS (Hypertext Transfer Protocol Secure) is the protocol which indicates  that the websiteis using the SSL/TLS. |
| Password less SSH | if you interact regularly with SSH commands and remote hosts, you may find that using a key  pair instead of passwords can be convenient. Instead of the remote system prompting for a  password with each connection, authentication can be automatically negotiated using a public  and private key pair. The private key remains secure on your own workstation, and the public  key gets placed in a specific location on each remote system that you access. A local caching  program such as ssh-agent or gnome-keyring allows you to enter that passphrase periodically,  instead of each time you use the key to access a remote system. if password authentication is  currently enabled, then the easiest way to transfer the public key to the remote host is with  the ssh-copy-id command . Popular asymmetric encryption algorithms are RSA,  Diffie-Hellman, ElGamal, and ECC |
| CA | **How does a company get a certificate?** Website owner first generates a public key and private key, keeping the private key secret. He  gives a *Certificate Signing Request file (CSR)* and his *public key* to the CA. CA then creates a personal certificate based on CSR including domain name, owner name,  expiry date, serial no. etc and also adds an **encrypted text** (= **digital signature**) to the  certificate and finally encrypts the whole certificate with the public key of the server and  sends it back to the website owner. This certificate is then decrypted with the private key of the website owner and finally,  he installs it on the website. Who are CA’s (Certificate Authorities)? CA’s are globally trusted companies like GoDaddy, GeoTrust, VeriSign etc  who provide digital certificates to the websites? |
| Microservices | The term “microservices” is generally meant to describe an approach to  software development that involves de-composing application functionality  into **individual components that can be deployed separately from**  each other, and typically communicate via application programming  interfaces or APIs.  Common characteristics of microservices:   * Support HTTP/REST protocols * Implement JSON or XML format for data exchange * Deployed via a containerization framework, such as Docker * Dynamically scaled on public or private cloud infrastructure * Often use noSQL or microSQL or key value stores to persist data  Cons of microservices As microservices heavily rely on messaging, they can face certain problems. |
| Hyperconvergence | Converged infrastructure involves a preconfigured package of software and  hardware in a single system for simplified management. |
| How ssl work | The client sends a "client hello" message. This includes the client's SSL version number,  cipher settings, session-specific data and other information that the server needs to  communicate with the client using SSL.  The server responds with a "server hello" message. This includes the server's SSL version  number, cipher settings, session-specific data, an SSL certificate with a public key and other  information that the client needs to communicate with the server over SSL.  The client verifies the server's SSL certificate from CA (Certificate Authority) and  authenticates the server.  Note SSL/TLS port is typically 443 .. Digital certificates is created by CA has metadata like  public key ,domain name expiry date |
| vmalloc | vmalloc allocates virtually contiguous memory space (not necessarily physically  contiguous), while kmalloc allocates physically contiguous memory (also virtually  contiguous). Most of the memory allocations in Linux kernel are done using kmalloc,  due to the following reasons:  On many architectures, hardware devices don’t understand virtual address. Therefore,  their device drivers can only allocate memory using kmalloc.  kmalloc has better  performance in most cases because physically contiguous memory region is more  efficient than virtually contiguous memory. interval of time. |
| Spinlock | Spinlock are really useful on SMP processor although a uniprocessor workstation  running a preemptive kernel behaves like SMP If a non-preemptive uniprocessor system  ever went into a spin on a lock, it would spin forever; no other thread would ever be  able to obtain the CPU to release the lock. spinlocks may be used in code that cannot  sleep, such as interrupt handlers and are acquired for short duration |
| Kernel mode | Kernel mode  -----------  Enter using interupt/Trap  1. Access to privileged instructions  --> CPU control instructions (CLI, STI, HLT, WAIT, LOCK, ...)  --> IN, OUT (direct hardware access)  2. Full access to physical memory (RAM)  User mode  ----------  1. Restricted instruction set  2. No direct hardware access  3. No access to entire physical memory (RAM)  4. Memory access only by virtual addresses (Virtual memory)  5. Memory access can happen via demand-paging |
| How system call works | How system call works   1. Application program makes a system call by invoking wrapper function in C library 2. This wrapper functions makes sure that all the system call arguments are available   to trap-handling routine   1. Generally a stack is used to pass these arguments to wrapper function. But the   Kernel looks into specific registers for these arguments. Hence the wrapper  function also takes care of copying these arguments to specific registers   1. Each system call has a unique call number which is used by kernel to identify   which system call is invoked. The wrapper function again copies the system call  number into specific CPU registers   1. Now the wrapper function executes trap instruction (int 0x80). This instruction   causes the processor to switch from 'User Mode' to 'Kernel Mode'   1. The code pointed out by location 0x80 is executed (Most modern machines use   sysenter rather than 0x80 trap instruction)   1. In response to trap to location 0x80, kernel invokes system\_call() routine which is   located in assembler file arch/i386/entry.S (also called handler)   1. This handler saves register values onto kernel stack and does some validations like   verifying system call number etc.   1. A map of system call number as key and the appropriate system call as value exists. This is called system\_call\_table. The handler uses this table to invoke appropriate system call service routine. It also validates the arguments if present. 2. After proper validations, the service routine performs required actions like modify   values at addresses specified in arguments or transfer data between user memory and kernel memory. After all these actions, service routine returns status of execution to the system\_call routine   1. Now the handler restores register values from kernel stack and places the system call return value on the stack 2. Thus handler is returned to wrapper function, simultaneously returning processor to user mode 3. Just in case if the return value of system call service routine indicated an error, then wrapper function sets 'errno' a global variable and then returns to caller providing integer return value that indicates the status of execution |
| copy\_from\_user | There are plenty of system calls (I/O operations such as write() are the  obvious examples) which require “reading” a chunk of data from user space so  the kernel can use it. In a virtual memory environment, there’s no guarantee  that the whole block of memory that you pass to write() is actually in RAM  at the time. It (or part of it) could be a memory-mapped file, or it (or part of it)  could be paged out. Or there are more exotic possibilities, such as that this  could be a NUMA architecture and some of the data is on another motherboard  . Or if [zswap](https://en.wikipedia.org/wiki/Zswap" \t "_blank" \o "en.wikipedia.org) is enabled, some of the data could be compressed. You get the idea. |

|  |  |
| --- | --- |
| Linux booting | **Stage 1**  When a system is booted, Processor executed a code from a well-known  location known as BIOS (Basic Input Output System) which is stored in flash  memory of motherboard. Its Job is to find the boot device(floppy/hard disk, cd).  When boot device is detected it passes control to first stage bootloader  **Stage 2**  FIRST stage boot loader is loaded into the RAM and executed. This Boot  Loader is 512 bytes in size with 64 bytes partition table). Its job is to find the  SECOND order Boot Loader (grub) and load it into RAM and passed control to  2nd stage bootloader.  **Stage 3**  Grub1 is is embedded in an MBR (size issue).  Grub2 is knowledge about the Linux file system (ext2,ext3)  Grub2 copy the Linux kernel image into the RAM using /boot/grub/grub.conf..  **Step 4 Kernel stage**  Kernel is in compressed cpio format file present in /boot directory .  Mounts the root file system as specified in the “root=” in grub.conf  grub> root (hd0,0)  grub> kernel /vmlinuz-i686-up-4GB root=/dev/hda9  grub> boot  kernel /vmlinuz-i686-up-4GB root=/dev/hda9 - Specifies the kernel location  which is inside the /boot folder. This location is related to the root(hd0,0)  statement.The root partition is specified according to the Linux naming  convention (/dev/hda9/)  initrd/ Initramfs is used by kernel as temporary root file system until kernel is booted  and thereal root file system is mounted. It also contains necessary drivers compiled  inside,which helps it to access the hard drive partitions, and other hardware.  insmod for loading kernel modules, and lvm (logical volume manager tools).  Initramfs/intrd is an image file in /boot containing the basic root file system with all  Kernel modules. The kernel then Mount this image file as a starting memory-based  root file system. The kernel then starts to detect the system’s hardware. The root file  system on disk takes over from the memory. The boot process then starts  INIT (SYSTEMD)  **Step 5 INIT**  The kernel, once it is loaded in step 4, it finds init in sbin (/sbin/init) and  executes it. The first thing init does is reading the initialization file, /etc/inittab.  The program **init** is the process with **process ID** 1. |
| Linux booting facts | Hot plug is the addition of a component to a running computer system without  significant interruption to the operation of the system. Hot plugging a device  does not require a restart of the system. |
| Device tree | Many embedded architectures have a lot of non-discoverable hardware.  Amongst the non-discoverable devices, a huge family are the devices that are  directly part of a system-on-chip: UART controllers, Ethernet controllers, SPI or  I2C controllers, graphic or audio devices, etc.  However, we still want all of these devices to be part of the device model.  Depending on the architecture, such hardware is either described in BIOS ACPI  tables (x86), using C code directly within the kernel, or using a special hardware  description language in a Device Tree.  Device Tree is a data structure for describing the hardware. This is where specific  information about the hardware is conveyed. It can be used to avoid hard coding  of every detail of a device into an Operating System. The device tree is passed to  the kernel at boot time. |
| UBOOT | U-Boot is both a first-stage and second-stage bootloader. If there are size  constraints, U-Boot may be split into stages:  U-Boot performs both first stage  (e.g., configuring memory controllers and SDRAM) and second-stage booting. ,  U-Boot requires its boot commands to explicitly specify the physical memory  addresses as destinations for copying data (kernel, ramdisk, device tree, etc.)  and for jumping to the kernel and as arguments for the kernel |
| Digital Signing | Digital Signature is a process that guarantees that the contents of a message  Have not been altered in transit. When you, the server, digitally sign a  document, you add a one-way hash (encryption) of the message content using  your private key . The recipient will recreate the message hash, decrypts the  encrypted hash using your well-known public key stored in your signed certificate,  check that both hashes are equals and finally check the certificate. |
| Hardware Security | Secure Boot is a security standard developed by members of the PC industry to help make surethat your PC boots using only software that is trusted by the PC manufacturer. When the PCstarts, the Bios checks the signature of each piece of boot software, including drivers and theoperating system. If the signatures are good, the PC boots, and the Bios gives control to theoperating system or else it would halt the boot up process and thrown an error. A Trusted Platform Module (TPM) is a hardware chip on the computer’s motherboard that  stores cryptographic keys used for encryption.Once enabled, the Trusted Platform  Module provides full disk encryption capabilities. It becomes the "root of trust" for the system  to provide integrity and authentication to the boot process. It keeps hard drives locked/sealed  until the system completes a system verification, or authentication check. The TPM includes a  unique RSA key burned into it, which is used for asymmetric encryption. Additionally, it can  generate, store, and protect other keys used in the encryption and decryption process.  A hardware security module (HSM) are external devices connected to a network using TCP/IP.  encryption capabilities by storing and using RSA keys. |
| Python | 1. Normal recommendations is to use 4 spaces 2. Python 3, the print statement has been replaced with a print () function. 3. lists are defined by having quoted values comma separated inside square brackets [ ] . 4. In Python dictionaries are written with curly brackets, and they are having quoted keys   and quoted values seprated by colons   1. the items of a dictionary by referring to its key name, inside square brackets:   Writing user-defined functions in Python. ...  Declare the function with the keyword def followed by function name arguments and colon.   1. def function (argument) colon 2. add 4 space and do action 3. return;   A lambda function is a small anonymous function.  A lambda function can take any number of arguments but can only have one expression.  Example vishal = lambda a comma b colon a plus b  print(vishal 20 comma 30))  Variables are containers for storing data values x=5  To create a class, use the keyword class: RegEx in Python import re  x = re.search (pattern ,input) |
| Dijkstra's Algorithm? | Dijkstra's algorithm is a step-by-step process we can use to find the shortest path between two  vertices in a weighted graph **.**   1. Pick first node and calculate distances to adjacent nodes. 2. Pick next node with minimal distance; repeat adjacent node distance calculations. 3. Final result of shortest-path tree.   What are advantage and disadvantage of Dijkstra's Algorithm?  1) It is used in Google Maps  2) ospf  Disadvantages: -  The major disadvantage of the algorithm is the fact that it does a blind  search there by consuming a lot of time waste of necessary resources.  Another disadvantage is that it cannot handle negative edges. This leads to  acyclic graphs and most often cannot obtain the right shortest path |
| **Apache Kafka** | Kafka is a distributed messaging system providing fast, highly scalable and redundant  Messaging through a publish-subscribe model. Kafka’s distributed design gives it several  advantages. First,Kafka allows many permanent or ad-hoc consumers. Second, Kafka is  highly available and resilient to node failures and supports automatic recovery.  All incoming data is first placed in Kafka and all outgoing data is read from Kafka.  Kafka is developed at LinkedIn and later it became a part of Apache Project  The most important elements of Kafka are as follows:  Topic is a category or feed name to which records are published. Topics in Kafka are always  multi-subscriber; i.e., a topic can have zero, one, or many consumers that can subscribe to the  topic and consume the data written to it.  You have sales records getting published in sales topic, product records getting published in  product topic and so on This will actually segregate your messages and the consumers will  only subscribe to the topic which they need.  Kafka topics are divided into a number of partitions. Partitions allow you to parallelize a topic by splitting the data in a particular topic across multiple brokers — each partition can be placed on a separate machine to allow multiple consumers to read from a topic parallelly.  So, in case of sales topic, you can have 3 partitions, from where three consumers can read data  parallelly. Producers publishes the data to the topics of their choice.  Consumers can subscribe to one or more topic and consumes data from that topic. Consumers  label themselves with a consumer group name. Each record published to a topic is delivered to  one consumer instance within each subscribing consumer group.  But you can have multiple consumer groups which can subscribe to a topic, where one record  can be consumed by as multiple consumers, i.e. one consumer from each consumer group.  Consumer instances can be in separate processes or on separate machines.  Broker is a single machine in the Kafka Cluster. Zookeeper is another apache open source project. It stores the metadata  information related to Kafka cluster like brokers information, topics details etc |
| Artificial intelligence | In Supervised learning, you train the machine using data which is well "labeled. .Supervised  learning model takes direct feedback to check if it is predicting correct output or not.  Regression: This type can be used when the output variable is a real or continuous value.  For example, salary based on work experience, or weight based on height, etc.  Classification:  To find whether an email received is a spam or ham  To identify customer segments  To find if a bank loan is granted  To identify if a kid will pass or fail in an examination   Decision Tree: find out if a person is fit or not. Based on a series of test conditions,Random Forest : A random forest, is making a bunch of decision trees, and consulting all of them at once to make your decision. Let's say you had to make a big life decision, and you  knew it came down to some factors: Health, happiness, finances, philosophy  And different factors have different importance’s:  Naive Bayes : assumes that the presence of a particular feature in a class is unrelated to  the presence of any other feature. For example, a fruit may be considered to be an  apple if it is red, round, and about 3 inches in diameter.  Unsupervised learning is where you only have input data and no corresponding output  Variable .The machine works by searching a pattern in the unlabeled data and  then responds. The goal of unsupervised learning is to find the hidden patterns and useful  insights from the unknown dataset.  Clustering: In clustering, objects are divided into clusters that are similar but are dissimilar to  the objects belonging to another cluster. Like, finding out which customers made similar  product purchases.  Association: As the name suggests, it discovers the probability of the co-occurrence of items in a  Collection. |
| Amazon Web Services (AWS) | Amazon Web Services (AWS) is a secure cloud services platform, offering compute  power, database storage, content delivery and other functionality to help businesses  scale and grow.  Basic Terminologies   1. Region — A region is a geographical area. Each region consists of 2 (or more)   availability zones.  2.  Availability Zone — It is simply a data center.  3. **IAM (Identity and Access Management)** — Allows you to manage users, assign  policies, create groups to manage multiple users.  Compute   1. EC2 (Elastic Compute Cloud) —  These are just the virtual machines in the   Cloud on which you have the OS level control. You can run whatever you  want in them.   1. ECS (Elastic Container Service) — It is a highly scalable container service to   allows You to run Docker containers in the cloud.   1. Lambda — AWS’s serverless technology that allows you to run functions in   the cloud.   1. Elastic Beanstalk — Allows automated deployment and provisioning of   resources like a highly scalable production website  storage  **S3 (Simple Storage Service) —** it uses Buckets (folders) and keys (files).  Storage service of AWS in which we can store objects like files, folders, images, documents,  Amazon Web Service S3 is basically a simple storage service that stores all the images, files,  data etc. Just like folders and files in Windows .  **Glacier** — It is an extremely low-cost archival service to store files for a long time like a few years or even decades.  **Storage Gateway** — It is a virtual machine that you install on your on-premise servers.  Your on-premise data can be backed up to AWS providing more durability  **CloudSearch** — It can be used to create a fully managed **search engine** for your website.  **Amazon Virtual Private Cloud (Amazon VPC) lets** you provision a logically isolated section of  the AWS Cloud where you can launch AWS resources in a virtual network that you define.  The public cloud provider is responsible for ensuring that each private cloud customer's data  remains isolated from every other customer's data both in transit and inside the cloud provider's  network. This can be accomplished through the use of security policies requiring some -- or  all -- of the following elements: encryption, tunneling, private IP addressing or allocating  a unique VLAN to each customer.  For example, you can create a public-facing subnet for your web servers that have access to the  internet. You can also place your backend systems, such as databases or application servers, in a private facing  subnet with no internet access. |
| IGMP snooping | IGMP snooping is the process of listening to Internet Group Management Protocol (IGMP)  network traffic. The feature allows a network switch to listen in on the IGMP conversation  between hosts and routers. By listening to these conversations, the switch maintains a map of  which links need which IP multicast streams. Multicasts may be filtered from the links which  do not need them and thus controls which ports receive specific multicast traffic.  The benefit is that multicast traffic goes only where it belongs. If nothing on that port has joined  that group, the switch will prune it. In a cascaded environment, that can significantly reduce  inter-switch traffic. And likewise keep access links clear of traffic hosts don't want. |
| DNS server | DNS is organized in a hierarchy that helps keep things running quickly and smoothly.   DNS information is shared among many servers but is also cached locally on client computers.  The initial request for the IP address is made to a recursive resolver, a server that is usually  operated by an ISP or other third-party provider. The recursive resolver knows which other  DNS servers it needs to ask to resolve the name of a site (networkworld.com) with its IP  address. This search leads to a root server, which knows all the information about top-level  domains, such as .com, .net, .org and all of those country domains like .cn (China)  and .uk (United Kingdom). Root servers are located all around the world, so the system  usually directs you to the closest one geographically. Once the request reaches the correct root  server, it goes to a top-level domain (TLD) name server, which stores the information for the  second-level domain, the words used before you get to the .com, .org, .net (for example, that  information for networkworld.com is “networkworld”). The request then goes to the Domain  Name Server, which holds the information about the site and its IP address. Once the IP address  is discovered, it is sent back to the client, which can now use it to visit the website. All of this  takes mere milliseconds. |
| (DNS) spoofing | Domain Name Server (DNS) spoofing (a.k.a. DNS cache poisoning) is an attack in which altered  DNS records are used to redirect online traffic to a fraudulent website that resembles its  intended destination.  [DNSSEC](https://www.imperva.com/learn/application-security/dnssec/) is a protocol designed to secure your DNS by adding additional methods of  verification.The protocol creates a unique cryptographic signature stored alongside your  other DNS recordsThis signature is then used by your DNS resolver to authenticate a  DNS response, ensuring that the record wasn’t tampered with. |
| PSIRT | The Cisco Product Security Incident Response Team (PSIRT) is a dedicated, global  team that manages the receipt, investigation, and public reporting of security  vulnerability information that is related Cisco products and networks. |
| CVE | CVE, short for Common Vulnerabilities and Exposures |
| Yocto Project | A Linux Foundation project that acts as an umbrella for various efforts to improve  Embedded Linux.  Yocto is a framework to create your own Linux distribution on a specific target hardware.  It is part of Linux Foundation . Releases on a 6-month cadence . yocto thud was release  in oct 2018  BitBake: A tool that reads metadata and runs tasks.  FOSS :Any open-source applications/libraries .Free and Open-Source Software(FOSS)  includes everything that goes into the SDK i.e sysroot,toolchain and Rootfs .Roofs comprise  the list of Open-Source applications including gdb, ip utilities. DNF replaces  Yum(runtime) package managers  Kernel is NOT PART of FOSS even though kernel sources are used to build FOSS  the FOSS packages would not include third-party vendor code(Broadcom, Marvell, Inphi,  etc) Having said that there are a couple of exceptions to this like ciscossh, ciscossl which  may move to FOSS eventually  XE is currently based off of Yocto Thud, and they sometimes add patches or  bring in newer versions of applications, and then we're based off of XE's XELinux  As of right now, we have a snapshot of their layers in our gitlab and are using that. Some of  the layers XELinux uses come from yocto, and XELinux may have made changes on top of  those layers, which we're using No changes have been directly made to any of the layers  coming from them. If any changes need to be made to the recipes, then there's a bbappend file in the meta-nx-linux  layer, which we created and control. Layer information is on the build wiki page, and are  locally cloned into our gitlab group.- |
| Yocto build | Metadata is any data that describes other data. Document metadata gives information  About a document such as the author, when it was created, when it was last modified,  and its size.  Image for post  ishagu2@yocto-nxos-bld04:~/nx-linux-dev\_new/build$ bitbake-layers show-layers  NOTE: Starting bitbake server...  layer path priority  ==========================================================================  meta /home/vishagu2/nx-linux-dev\_new/distro/openembedded-core/../openembedded-core/meta 5  meta-oe /home/vishagu2/nx-linux-dev\_new/distro/openembedded-core/../meta-openembedded/meta-oe 6  meta-perl /home/vishagu2/nx-linux-dev\_new/distro/openembedded-core/../meta-openembedded/meta-perl 6  meta-python /home/vishagu2/nx-linux-dev\_new/distro/openembedded-core/../meta-openembedded/meta-python 7  meta-networking /home/vishagu2/nx-linux-dev\_new/distro/openembedded-core/../meta-openembedded/meta-networking 5  meta-filesystems /home/vishagu2/nx-linux-dev\_new/distro/openembedded-core/../meta-openembedded/meta-filesystems 6  meta-webserver /home/vishagu2/nx-linux-dev\_new/distro/openembedded-core/../meta-openembedded/meta-webserver 6  meta-virtualization /home/vishagu2/nx-linux-dev\_new/distro/openembedded-core/../meta-virtualization 8  meta-security /home/vishagu2/nx-linux-dev\_new/distro/openembedded-core/../meta-security 8  meta-rhel-host /home/vishagu2/nx-linux-dev\_new/distro/openembedded-core/../meta-rhel-host 5  meta-nx-linux /home/vishagu2/nx-linux-dev\_new/distro/openembedded-core/../meta-nx-linux 50  Yocto provides the environment for compiling all the packages that are required to  boot a board . It works on the meta layers. Under the meta layers, there are different  different recipes for each package. Under these recipes, there are .bb files for each  package. During compilation, these .bb files are used to get all the information about a  particular package. This. files contain information like the License, URL to download  the source code, what are the flags should pass at the time of configuration or compilation.  Yocto terms  1>Poky is the name of build system used by yocto  2>Bit bake is a Powerful and flexible python-based build engine  3>Layer: A collection of recipes .Typically, each layer is organized around a specific  theme,  e.g. adding recipes for building web browser software. Open Embedded-Core is a base  layer of recipes, classes and associated files that is meant  4> All the artefacts generated are stored in the deploy folder  4>Metadata – Metadata is collection of below items   1. Configuration (\*.conf) :Drives the overall behavior of the build process 2. Recipes (\*.bb) : Usually describe build instructions for a single package 3. Append files (\*.bbappend) :Can add or override previously set values 4. Classes (\*.bbclass) :Inheritance mechanism for common functionality .   bbclass) are used to factorize recipe's code, to handle some general problems.  For instance, handling example inherit logging  Recipe script :  1>Locate and download source code ,  2>Unpack source into working directory  3>Apply any patches Perform any necessary pre-build configuration  4>Compile the source code  5>Installation of resulting build artifacts in WORKDIR  6>Copy artifacts to sysroot ,Create binary package(s)  We have a bbappend file that supplies a set of patches. It currently has the unintended side-effect of patching both the native version used during the Yocto build process, and the eventual target version. How do I modify the recipe such that it only acts upon the target version?  SRC\_URI\_append\_class-target = " file://..."  SRC\_URI Where to obtain the upstream sources and which patches to apply (this is called  “fetching”)  bitbake-layers create-layer ->Use a new custom layer for modularity and maintainability.  They all start with “meta-” by convention  tmp/log/cooker will have all logs |
| NX-Linux (NXL) | NXL is a Linux Distribution for NXOS based based on Xilinx distro(Yocto Thud) with  NXOS customizations including GCC 5.2 and and Clang 7.0  Following git repository mirrors are setup from XE sources - meta-open embedded,  meta-virtualization, meta-security, scripts, meta-nx-linux (Specific to NXOS for FOSS  customizations  A sysroot is a directory which is the root directory for the purpose of locating headers  and libraries. You need sysroot for doing cross compilation and building toolchain |
| Switch words | **Supervisor engine**: it is basically the control plane .  **Line card**: data plane/responsible for packet forwarding  **Fabric module**:It interconnects two different line-cards also supervisor card of the switch.  **Fabric extender:** a line card connected using fabric has no capability to store a forwarding table or run any control  plane protocols |
| **Bitbake Interview**  **Questions** | BitBake is a program written in the Python language. At the highest level, BitBake  interprets metadata, decides what tasks are required to run, and executes those tasks.  Similar to GNU Make, BitBake controls how software is built. GNU Make achieves its  control through "makefiles". BitBake uses "recipes".  BitBake Recipes, which are denoted by the file extension .bb, are the most basic metadata  files. These recipe files provide BitBake with the following: version,• Existing Dependencies  how to compile .  Class files(.bbclass) extension, contain information that is useful to share between metadata  files. The BitBake source tree currently comes with one class metadata file called  base.bbclass. You can find this file in the classes directory. The base.bbclass is special since  it is always included automatically for all recipes and classes. This class contains definitions  for standard basic tasks such as fetching, unpacking, configuring (empty by default),  compiling (runs any Makefile present), installing (empty by default) and  packaging (empty by default). These tasks are often overridden or extended by  other classes added during the project development process.  Layers allow you to isolate different types of customizations from each other  Append files, which are files that have the .bbappend file extension, add or  extend build information to an existing recipe file.  busybox\_1.21.%.bbappend That append file would match any busybox\_1.21.x.bb  version of the recipe. So, the append file would match the following recipe  names: busybox\_1.21.1.bb busybox\_1.21.2.bb busybox\_1.21.3.bb |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |