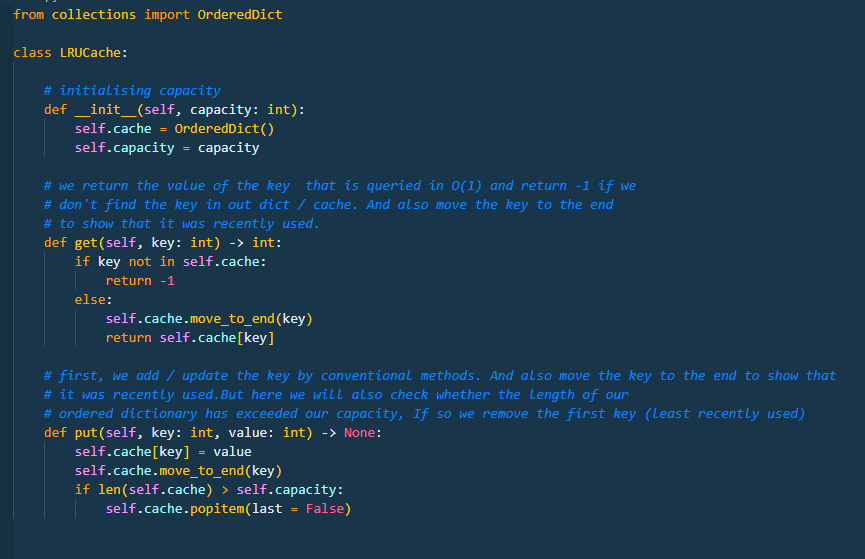
**Q1. Write an implementation LRU cache in python.**

Answer:

LRU (Least Recently Used) Cache discards the least recently used items first. This algorithm requires keeping track of what was used when, which is expensive if one wants to make sure the algorithm always discards the least recently used item. General implementations of this technique require keeping “age bits” for cache-lines and track the “Least Recently Used” cache-line based on age-bits.



**Q2.  What happens when you type www.attainu.com in your browser?**

Answer:

Let's understand the different actions that are being perform by network and browser when we enter www.attainu.com:

1. First and foremost thing that happen is that the browser checks the cache for a DNS record to find the corresponding IP address of www.attainu.com. The browser maintains a repository of DNS records for a fixed duration for websites you have previously visited. So, it is the first place to run a DNS query. DNS(Domain Name System) is a database that maintains the name of the website (URL) and the particular IP address it links to

2. The browser will now check the OS cache. If it is not in the browser cache, the browser will make a system call (i.e., gethostname on Windows) to your underlying computer OS to fetch the record since the OS also maintains a cache of DNS records. If it’s not on your computer, the browser will communicate with the router that maintains its’ own cache of DNS records.If all steps fail, the browser will move on to the ISP. Your ISP maintains its’ own DNS server, which includes a cache of DNS records, which the browser would check with the last hope of finding your requested URL.

3. If the requested URL is not in the cache, ISP’s DNS server initiates a DNS query to find the IP address of the server that hosts www.attainu.com.The purpose of a DNS query is to search multiple DNS servers on the internet until it finds the correct IP address for the website. This type of search is called a recursive search since the search will repeatedly continue from a DNS server to a DNS server until it either finds the IP address we need or returns an error response saying it was unable to find it.

4. Once the browser receives the correct IP address, it will build a connection with the server that matches the IP address to transfer information. Browsers use internet protocols to build such connections. There are several different internet protocols that can be used, but TCP is the most common protocol used for many types of HTTP requests.

5. Once the TCP connection is established, it is time to start transferring data! The browser will send a GET request asking for www.attainu.com web page. If you’re entering credentials or submitting a form, this could be a POST request. It will also pass information taken from cookies the browser has in store for this domain.

6. The server contains a webserver that receives the request from the browser and passes it to a request handler to read and generate a response. The request handler is a program that reads the request, its’ headers, and cookies to check what is being requested and also update the information on the server if needed. Then it will assemble a response in a particular format (JSON, XML, HTML).

7. The server response contains the web page you requested as well as the status code, compression type (Content-Encoding), how to cache the page (Cache-Control), any cookies to set, privacy information, etc.

8. The browser displays the HTML content in phases. First, it will render the bare bone HTML skeleton. Then it will check the HTML tags and send out GET requests for additional elements on the web page, such as images, CSS stylesheets, JavaScript files, etc. These static files are cached by the browser, so it doesn’t have to fetch them again the next time you visit the page. In the end, you’ll see www.attainu.com appearing on your browser.

**Q3. What is the difference between private and public ip?**

Answer:

Private IP Address and Public IP Address are used to uniquely identify a machine on the internet. Private IP address is used with a local network and public IP address is used outside the network. Public IP address is provided by ISP, Internet Service Provider.

Following are the important differences between Private IP Address and Public IP Address.

| Sr. No. | Key | Private IP Address | Public IP Address |
| --- | --- | --- | --- |
| 1 | Scope | Private IP address scope is local to present network. | Public IP address scope is global. |
| 2 | Communication | Private IP Address is used to communicate within the network. | Public IP Address is used to communicate outside the network. |
| 3 | Format | Private IP Addresses differ in a uniform manner. | Public IP Addresses differ in varying range. |
| 4 | Provider | Local Network Operator creates private IP addresses using network operating system. | ISP, Internet Service Provider controls the public IP address. |
| 5 | Cost | Private IP Addresses are free of cost. | Public IP Address comes with a cost. |
| 6 | Locate | Private IP Address can be located using ipconfig command. | Public IP Address needs to be searched on search engine like google. |
| 7 | Range | Private IP Address range:  10.0.0.0 – 10.255.255.255,  172.16.0.0 – 172.31.255.255,  192.168.0.0 – 192.168.255.255 | Except private IP Addresses, rest IP addresses are public. |
| 8 | Example | Private IP Address is like 192.168.11.50. | Public IP Address is like 17.5.7.8. |

**Q4. Describe deadlock characteristics.**

Answer:

In Computer Science a set of process is said to be in deadlock if each process in the set is waiting for an event that only another process in the set can cause. Since all the processes are waiting, none of them will ever cause any of the events that would wake up any of the other members of the set & all the processes continue to wait forever.

Following are the condition for deadlock:

1. Mutual exclusion

Only one process at a time can use a resource.

2. Hold and wait

Process holding at least one resource is waiting to acquire additional resources held by other

processes.

3. No pre-emption

Resources are released only voluntarily by the process holding the resource, after the process is finished with it.

4. Circular wait

There exists a set {P1 , …, Pn } of waiting processes.

P1 is waiting for a resource that is held by P2

P2 is waiting for a resource that is held by P3

…

Pn is waiting for a resource that is held by P1.

**5. Explain in brief -**

**1. Swap memory.**

**2. Context switching**

**3. IPv4 address**

**4. IPV6 address**

Answer:

1. Swap Memory:

For every operating system, there is a dedicated amount of RAM available that makes the processing of a program possible. However, the amount of this RAM is limited which is why RAM cannot hold a bulk of data in it. Therefore, there should be a backup option available which can support RAM whenever it runs out of memory.

This concept holds for the Windows operating system as well as for Linux. In Windows OS, whenever RAM has an insufficient amount of memory to hold a process, it borrows some amount of memory from the secondary storage. This borrowed memory is known as Virtual Memory. Similarly, whenever RAM runs out of memory in Linux, it borrows some memory from the secondary storage to store its inactive content.

In this way, the RAM finds sufficient space to hold a new process within it. Here, the borrowed space from the hard disk is called Swap Memory. In this article, we will try to learn the concept of swap memory in detail.

1. Context switching:

Context Switching involves storing the context or state of a process so that it can be reloaded when required and execution can be resumed from the same point as earlier. This is a feature of a multitasking operating system and allows a single CPU to be shared by multiple processes.

There are three major triggers for context switching. These are given as follows −

* **Multitasking**: In a multitasking environment, a process is switched out of the CPU so another process can be run. The state of the old process is saved and the state of the new process is loaded. On a pre-emptive system, processes may be switched out by the scheduler.
* **Interrupt Handling**: The hardware switches a part of the context when an interrupt occurs. This happens automatically. Only some of the context is changed to minimize the time required to handle the interrupt.
* **User and Kernel Mode Switching**: A context switch may take place when a transition between the user mode and kernel mode is required in the operating system.

1. IPv4 address:

The IPv4 address is a 32-bit number that uniquely identifies a network interface on a system, as explained in [How IP Addresses Apply to Network Interfaces](https://docs.oracle.com/cd/E23823_01/html/816-4554/ipplan-5.html#ipplan-25). An IPv4 address is written in decimal digits, divided into four 8-bit fields that are separated by periods. Each 8-bit field represents a byte of the IPv4 address. This form of representing the bytes of an IPv4 address is often referred to as the dotted-decimal format.

The following figure shows the component parts of an IPv4 address, 172.16.50.56.

IPv4 Address Format:

image:The figure divides the IPv4 address into two parts, network part and network host, which are described in the next context.

**172.16**

Registered IPv4 network number. In class-based IPv4 notation, this number also defines the IP network class, Class B in this example that would have been registered by the IANA.

**50.56**

Host part of the IPv4 address. The host part uniquely identifies an interface on a system on a network. Note that for each interface on a local network, the network part of the address is the same, but the host part must be different.

Each IPv4-based network must have the following:

* A unique network number that is assigned by either an ISP, an IR, or, for older networks, registered by the IANA. If you plan to use private addresses, the network numbers you devise must be unique within your organization.
* Unique IPv4 addresses for the interfaces of every system on the network.
* A network mask.

1. IPv6 address:

IPv6 is the successor to the first addressing infrastructure of the [Internet](https://en.wikipedia.org/wiki/Internet), [Internet Protocol version 4](https://en.wikipedia.org/wiki/Internet_Protocol_version_4) (IPv4). In contrast to IPv4, which defined an IP address as a 32-bit value, IPv6 addresses have a size of 128 bits. Therefore, in comparison, IPv6 has a vastly enlarged [address space](https://en.wikipedia.org/wiki/Address_space). IPv6 does not support broadcast addresses, but instead uses multicast addresses for broadcast. In addition, IPv6 defines a new type of address called anycast.

IPv6 Address Representation

IPv6 addresses consist of 8 groups of 16-bit hexadecimal values separated by colons (:). IPv6 addresses have the following format:

**aaaa:aaaa:aaaa:aaaa:aaaa:aaaa:aaaa:aaaa**

Each **aaaa** is a 16-bit hexadecimal value, and each **a** is a 4-bit hexadecimal value. Following is a sample IPv6 address:

**3FFE:0000:0000:0001:0200:F8FF:FE75:50DF**

You can omit the leading zeros of each 16-bit group, as follows:

**3FFE:0:0:1:200:F8FF:FE75:50DF**

You can compress 16-bit groups of zeros to double colons (::) as shown in the following example, but only once per address:

**3FFE::1:200:F8FF:FE75:50DF**

An IPv6 address prefix is a combination of an IPv6 prefix (address) and a prefix length. The prefix takes the form ipv6-prefix/prefix-length and represents a block of address space (or a network). The ipv6-prefix variable follows general IPv6 addressing rules. The /prefix-length variable is a decimal value that indicates the number of contiguous, higher-order bits of the address that make up the network portion of the address. For example, 10FA:6604:8136:6502::/64 is a possible IPv6 prefix.

For more information on the text representation of IPv6 addresses and address prefixes, see RFC 4291, IP Version 6 Addressing Architecture.

**IPv6 Address Types**

IPv6 has three types of addresses:

* Unicast—For a single interface.
* Multicast—For a set of interfaces on the same physical medium. A packet is sent to all interfaces associated with the address.
* Anycast—For a set of interfaces on different physical media. A packet is sent to only one of the interfaces associated with this address, not to all the interfaces.