In computer architecture, instruction prefetch is a technique used in central processor units to speed up the execution of a program by reducing wait states.

Prefetching occurs when a processor requests an instruction or data block from main memory before it is actually needed. Once the block comes back from memory, it is placed in a cache. When the instruction/data block is actually needed, it can be accessed much more quickly from the cache than if it had to make a request from memory.

​The browser (Chrome, Firefox, etc.) caches this content in the background, making it instantly available if the user clicks on a link that uses the content.

​Today’s websites are driven by dynamic content and multimedia. A single web page can require multiple images and videos to be cached, resulting in long wait times and a poor user experience.

​Prefetching resolves this issue by assuming which links the user is likely to click, then downloading the content of those links. If the user decides to click on one of the links, then the page will be rendered instantly as the content has already been downloaded.

​A typical domain lookup can take anywhere from 1ms to several seconds depending on whether the domain name is cached locally. DNS prefetching reduces domain lookup times by resolving the domain name in advance.

For example, Google Chrome implements DNS prefetching by scanning the hyperlinks in a page, extracting the domain names, and resolving them to IP addresses while the user browses the page. DNS prefetching is best used when there are multiple links that refer to external websites, such as the results in a search engine.

​

Prerendering is an extension of prefetching used by Chrome and Internet Explorer.

While prefetching downloads the content of a webpage, prerendering renders the entire page in the background. The prerendered page is treated as a separate tab, so when the user clicks on a prerendered link the current tab is essentially replaced by the prerendered tab.

In some cases, Chrome and Internet Explorer can analyze the content and layout of a website in order to perform automatic prerendering. For example, online periodicals can use prerendering to instantly serve the next article when the user is finished with the current article.

​Internet Explorer 11 introduced Page Prediction, which analyzes browsing habits and page content to perform automatic prefetching. When a user opens the page, Page Prediction scans for links that the user is likely to click on, such as page numbers in an article. The browser downloads the resources for those links in the background, letting the page load almost instantly when the user clicks on it.

---------

​​​Internet Explorer’s extended prefetching features include caching DNS queries, limiting prefetching for metered connections and disabling prefetching to conserve battery life. Browsers will always prioritize active downloads over prefetching and even pause prefetching if bandwidth is limited.

**Users** experience faster load times for web content as pages are downloaded in the background. There’s no interruption to the user’s current browsing experience.

**Administrators** see less spiked bandwidth usage since the content the user is likely to see has already been downloaded. Resources are delivered as needed, rather than all at once.

**Enterprises** see higher visitor retainment as visitors no longer need to wait to access a web page. The browsing experience feels like the website is being hosted locally.

**​With a bit of behavioral analysis and developer creativity, prefetching works to make the web faster.**