

توضیح اجمالی فرآیند:

فراخوانی باید به صورت نام پورت ، سیاست کش ، و اندازه کش انجام شود

`./proxy 1111 2 500`

دستور بالا به معنی فعال شدن پروکسی بر روی پورت 111 با سیاست دوم و با تا تعداد 500 عدد فایل در کش می‌باشد.

تابع fs نقش چاپ مقادیر با فرمت رنگی در صورت پشتیبانی ترمینال را دارد.

مدیریت فایل‌های کش شده توسط یه پردازش با نام فرضی cacheManager انجام می‌شود

در ابتدای برنامه یک پایپ pipe ایجاد شده سپس برنامه fork می‌شود و پردازش مدیریت کش در حلقه خواندن از روی پایپ قرار می‌گیرد، به علت باز بودن نوشتن بر روی پایپ ، دستور خواندن بلاک کننده بوده و سپس بر روی پورت مربوطه Listen می‌کند و با رسیدن هر درخواست یک Fork برای رسیدگی به درخواست و پورت جدید ایجاد می‌کند و ..

بعد از دریافت فایل ، یا ارسال آن به کلاینت به مدیرکش توسط پایپ اطلاع داده می‌شود تا در صورت لزوم بر اساس سیاست اتخاذ شده کش کردن ، فایل‌های اضافی را حذف کند.

فایل‌ها در واقع پاسخ سرور به درخواست get پروکسی می‌باشند ، که عملاً از هاش کد MD5 آدرس URL مربوطه نام آن‌ها ساخته می‌شود.

```
344
345     ss.str("");
346     ss << req->host << req->path;
347     fs(10, Spid + " host---path : =====>\r\n " + ss.str());
348     string fileName = md5(ss.str());
349     fs(14, Spid + "Md5Hash: " + fileName);
350
```

فایل‌ها درون شاخه /CACHE ذخیره می‌شوند

از یک قفل موتاکس استفاده شده تا در صورتی که دو درخواست یکسان همزمان به پروکسی رسید یکی از آن‌ها اقدام به دریافت فایل کند و درخواست دوم تا 5 ثانیه معطل می‌ماند.

```
int cond = 0;
{
    pthread_mutex_lock(&m);
    cond = (fileExistInCache(fileName)) ? 1 : cond;
    cond = fileExistInCache(fileName + "_Done") ? 2 : cond;
    if (cond == 0)
    {
        rFileName = "./CACHE/" + fileName;
        std::ofstream outfile("./CACHE/" + fileName, ios::out | ios::binary);
        rFileName = "./CACHE/" + fileName;
        outfile.open("./CACHE/" + fileName, ios::out);
        outfile.close();
    }
    pthread_mutex_unlock(&m);
}
```

```

if (cond == 1)
{
    fs(13, Spid + " CACHE HIT But Last Request Hasen't Done Yet Wait " + fileName);
    int lCount = 0;
    while ((fileExistInCache(fileName) == true) && (fileExistInCache(fileName + "_Done") == false) && (lCount++ < 5))
        sleep(1);
    cond = fileExistInCache(fileName + "_Done") ? 2 : 0;
}

```

در حین کش کردن یک فایل اطلاعات به سمت کلاینت نیز ارسال می‌شود و با پایان آن به نام فایل مقدار Done\_\_ اضافه می‌شود.

در صورتی که پاسخ سرور شامل HTTP/1.1 200 ok باشد فایل‌ها ذخیره خواهند شد. دو پیام Response is okay و Bad Response و مقداری از محتوای پاسخ نیز بر صفحه نمایش داده می‌شود.

```

217
218         if (strncmp(buf, "HTTP/1.1 200 OK", 15) == 0)
219         {
220             isResponseEqual200 = true;
221             fs(11, Spid + " Response is Okay");
222             fs(11, bufT0);
223         }
224         else
225         {
226             fs(12, Spid + " Bad Response");
227             fs(12, bufT0);
228         }
229     }

```

در صورت موفق بودن دریافت فایل به مدیرکش از طریق پایپ اطلاع داده می‌شود

```

251         if ((isResponseEqual200) && (iRecv == 0))
252         {
253             rename(relativeFileName.c_str(), (relativeFileName + "_Done").c_str());
254             write(fdCacheManager[1], (relativeFileName + "_Done").c_str(), MSGSIZE);
255         }

```

همچنین سیگنال 13 مربوط به ارسال بر روی سوکت بسته شده نیز مدیریت شده تا در صورت لغو عمل دریافت فایل توسط کلاینت در صورت امکان فایل نیمه کاره‌ی کش شده حذف شود

```

407 void signalHandler(int signum)
408 {
409     // cout << Spid<< " Interrupt signal (" << signum << ") received.\n";
410     if (signum == 13)
411     {
412         if (rFileName != "")
413         {
414             fs(18, "\t\tTrying To Remove Temp File " + rFileName + "\t\t", false);
415             remove(rFileName.c_str());
416         }
417         exit(signum);
418     }
419
420     if (signum == 2)
421     {
422         if (MainPrgPid == getpid())
423             close(MainSockfd);
424
425         exit(signum);
426     }
427 }

```

\*\*\* درخواست های رسیده از مرورگر (درخواست های get کلاینت) تغییر کرده و هدر آن تغییر می کند.

```

53
54     /* Set headers */
55     ParsedHeader_set(req, "Host", req->host);
56     ParsedHeader_set(req, "Connection", "close");
57     ParsedHeader_modify(req, "Cache-Control", "no-cache");
58     ParsedHeader_modify(req, "If-None-Match", "AA");
59     ParsedHeader_modify(req, "If-Modified-Since", "AA");
60     ParsedHeader_modify(req, "Pragma", "no-cache");

```

اطلاعات عدم کش به آن اضافه می شود تا سرور دوباره فایل را ارسال کند

همچنین در صورتی که درخواست get شامل Range باشد (مثلا در زمان دانلود یک فایل مرورگرها چندین کانکشن برای دریافت سریعتر فایل باز می کنند) آن درخواست لغو می شود

```

47
48     if (ParsedHeader_get(req, "Range") != NULL)
49     {
50         fs(12, "\t!!Due To Cache Complexity Pratial Request Is Not Allowed!!\t", false);
51         exit(1);
52     }

```

## Fork اول جهت ایجاد مدیرکش با توجه به سیاست مربوطه

```

pid_t p;
p = fork();

if (p < 0)
{
    fprintf(stderr, "fork Failed");
    return 1;
}
// child process
if (p == 0)
{
    list<fileAccessData> ls0;
    struct stat t_stat;

    const auto &directory_path = std::string("./CACHE");
    const auto &files = GetDirectoryFiles(directory_path);
    for (const auto &file : files)
    {
        fileAccessData t0;
        t0.fileName = "./CACHE/" + file;
        stat(t0.fileName.c_str(), &t_stat);
        struct tm *timeinfo = gmtime(&t_stat.st_ctime);
        t0.firstTimeCreated = (timeinfo->tm_sec+(timeinfo->tm_min*60)+(timeinfo->tm_hour*60*60)+(timeinfo->tm_mday*24*60*60));
        t0.lastTimeAccessed = timeSinceEpochMillisec();
        t0.numberOfTimeAccessed = 0;
        ls0.push_front(t0);
    }

    int missN=0, hitN=0;
    int nbytes;
    char inbuf[MSGSIZE];
    while (true)
    {
        fileAccessData t0;

        if ((missN + hitN) > 0)
        {
            fs(17, "Cache Manager " + Spid + "\t MissN: " + to_string(missN) + "\t HitN: " + to_string(hitN) + "\t\n");
            if ((nbytes = read(fdCacheManager[0], inbuf, MSGSIZE)) > 0)

```

ابتدا تمامی اطلاعات فایل‌های داخل کش با ساختمان داده مربوطه اضافه شده و زمان ساخت آن‌ها (این زمان جهت مقایسه می‌باشد در ادامه هم فرمت نیست اما مشکلی ایجاد نمی‌کند) نیز اضافه می‌شود.

## ساختمان داده مربوطه

```

18 struct fileAccessData
19 {
20     string fileName;
21     uint64_t lastTimeAccessed;
22     uint64_t numberOfTimeAccessed;
23     uint64_t firstTimeCreated; //time(&time1);
24     uint64_t getData(int cacheStrategy)
25     {
26         switch (cacheStrategy)
27         {
28             case 1:
29                 return firstTimeCreated;
30                 break;
31             case 2:
32                 return lastTimeAccessed;
33                 break;
34             default:
35                 return numberOfTimeAccessed;
36                 break;
37         }
38     }
39 }
40 };

```

اطلاعات مربوط به درصد cach hit & miss نیز بر صفحه چاپ می‌شود .

تصاویری از ترمینال و اجرا:

```
esi@esi-virtual-machine: ~/Desktop/CProjects/HTTP-PROXY
File Edit Tabs Help
esi@esi-virtual-machine:~/Desktop/CProjects/HTTP-PROXY$ ./proxy 1111
Provide A Port ! And Also Chaching Strategy and cacheMaximon Size
1:Fifo
2:LRU
3:LFU
./proxy <portnumber> <Chaching Strategy> <maxCacheSize>
esi@esi-virtual-machine:~/Desktop/CProjects/HTTP-PROXY$ ./proxy 1111 1 6
Welcome
```

درخواست اولیه و تغییر کرده و آدرس درخواستی:

```
esi@esi-virtual-machine:~/Desktop/CProjects/HTTP-PROXY$ ./proxy 1111 1 6
Welcome

pid: 10734 request_message: Orginal 1 =====>GET http://www.columbia.edu/~fdc/sample.html HTTP/1.1
Host: www.columbia.edu
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86 64; rv:65.0) Gecko/20100101 Firefox/65.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Cookie: BIGipServer-CUIT-www.columbia.edu-80-pool=1781021568.20480.0000
Upgrade-Insecure-Requests: 1

<=====

pid: 10734 browser req: Changed 1 =====>GET /~fdc/sample.html HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86 64; rv:65.0) Gecko/20100101 Firefox/65.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: BIGipServer-CUIT-www.columbia.edu-80-pool=1781021568.20480.0000
Upgrade-Insecure-Requests: 1
Host: www.columbia.edu
Connection: close

<=====

pid: 10734 host---path : =====>
www.columbia.edu/~fdc/sample.html
```



کد هش آدرس و دریافت اوکی بودن پاسخ و حجم فایل دریافتی و اطلاعات خاکستری رنگ توسط پایپ به مدیر کش رسیده و جزئیات میس و هیت

```
pid: 10734 host--path : =====>
www.columbia.edu/~fdc/sample.html

pid: 10734 Md5Hash: 03e91b1569fe4699e3a22f89fff91719

pid: 10734 CACHE MISS 03e91b1569fe4699e3a22f89fff91719

pid: 10734 Response is Okay

HTTP/1.1 200 OK
Date: Wed, 29 May 2019 12:13:03 GMT
Server: Apache
Last-Modified: Sun, 26 May 2019 16:49:01 GMT
Accept-Ranges: bytes
Vary: Accept-Encoding,User-Agent
Content-Encoding: gzip
C...

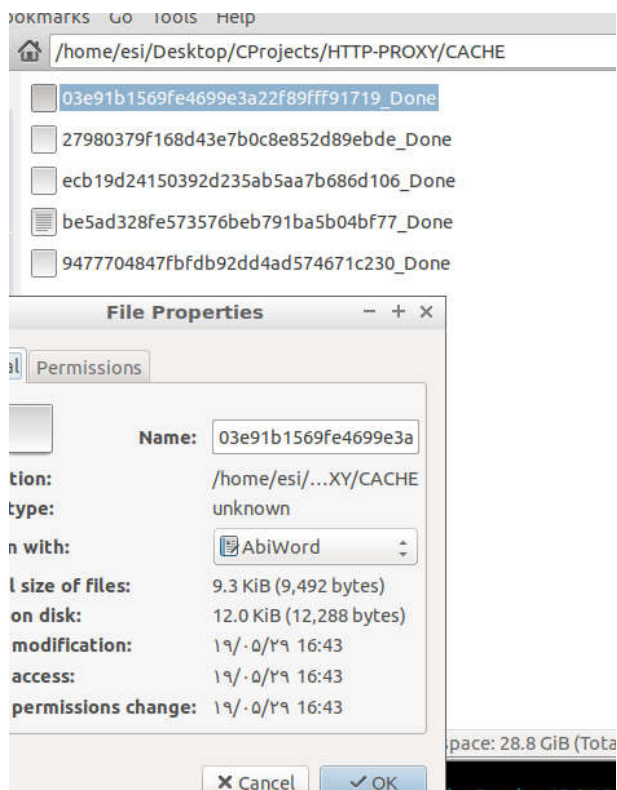
./CACHE/03e91b1569fe4699e3a22f89fff91719 pid: 10734 === done -- total totalRccv: 9492

Cache Manager MISS pid: 10377 ./CACHE/03e91b1569fe4699e3a22f89fff91719_Done
numberOfTimeAccessed: 1 lastTimeAccessed: 1559131983544 firstTimeCreated: 1559131983544

Cache Manager pid: 10377 MissN: 1 HitN: 0 MissRate: 100% HitRate: 0%

Cache Manager MISS pid: 10377 ./CACHE/03e91b1569fe4699e3a22f89fff91719_Done
numberOfTimeAccessed: 1 lastTimeAccessed: 1559131983544 firstTimeCreated: 1559131983544
```

همین فایل در شاخه مربوطه تولید شده است.



نمونه دوم درخواست فایل پی دی اف زیر و کنسل کردن درخواست در مرورگر

```
pid: 11235 host--path : =====>  
www.reelreality.com/files/Understanding_Binary-Hexadecimal.pdf
```

```
pid: 11235 Md5Hash: bbc16d7da96f30dc207dda6fb61bf028
```

```
pid: 11235 CACHE MISS bbc16d7da96f30dc207dda6fb61bf028
```

```
pid: 11235 Response is Okay
```

```
HTTP/1.1 200 OK
```

```
Content-Length: 458604
```

```
Content-Type: application/pdf
```

```
Last-Modified: Wed, 25 Oct 2006 18:36:57 GMT
```

```
Accept-Ranges: bytes
```

```
ETag: "366358d64f8c61:a4b"
```

```
Server: Microsoft-IIS/6.0
```

```
X-...
```

```
Trying To Remove Temp File ./CACHE/b8f58da96f302413aa609cf77388a87a4  
Trying To Remove Temp File ./CACHE/bbc16d7da96f30dc207dda6fb61bf028
```

```
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:65.0) Gecko/20100101 Firefox/65.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

<=====

```
pid: 11251 browser req: Changed 449 =====>GET /favicon.ico HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:65.0) Gecko/20100101 Firefox/65.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Host: www.reelreality.com
Connection: close
```

<=====

```
pid: 11251 host---path : =====>
www.reelreality.com/favicon.ico
```

```
pid: 11251 Md5Hash: b8f58da96782473aa609cf77388587a1
```

```
pid: 11251 CACHE MISS b8f58da96782473aa609cf77388587a1
```

```
pid: 11251 Bad Response
```

```
HTTP/1.1 404 Not Found
Content-Length: 1635
Content-Type: text/html
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Date: Wed, 29 May 2019 12:20:48 GMT
Connection: close
```

```
<!DOCTYPE HTML PUB...
```

```
!!Due To Cache Complexity Pratial Request Is Not Allowed!!
ratial Request Is Not Allowed!!
!!Due To Cache Complexity Pratial Request Is Not Allowed!!
!!Due To Cache Complexity Pratial Request Is Not Allowed!!
!!Due To Cache Complexity Pratial Request Is Not Allowed!!
```



در مورد تحلیل استراتژی کش کردن حالت‌های متفاوتی ممکن است رخ دهد برای مثال فرض کنید سیاست Least Recently use استفاده شده و حجم حافظه کش 5 فایل باشد و این 5 فایل 100 بار بارگذاری شده باشند ، حال سایت دیگری را باز کرده و آن نیاز به 5 فایل دیگر دارد و آن‌ها در کش موجود نباشند، در نتیجه دائماً همین فایل‌های فعلی از کش اخراج می‌شوند (حتی اگر 1 عدد هم بیشتر نگه داریم همین وضعیت است) و درمورد FIFO نیز همچنان ممکن است MissRate با افزایش تعداد فایل‌ها بیشتر شود

و ..

\* تعدادی دستور مربوط به اجرای فایل کامپایل شده بر روی اوبونتو 18.04 توسط داکر نیز در انتها وجود دارد . که مرتبط با فضای کاری داخل ایميج، کپی کردن فایل‌ها بر روی ایميج و در معرض قرار دادن پورت بین ایميج و محیط کاری و اجرای برنامه پروکسی .. هستند

test sites:

<http://www.columbia.edu/~fdc/sample.html>

<http://rebeca-lang.org/>

[http://www.reelreality.com/files/Understanding\\_Binary-Hexadecimal.pdf](http://www.reelreality.com/files/Understanding_Binary-Hexadecimal.pdf)