Project: A Http Web Server Based on C and Linux API

Name: wenbin huang

NetID: vx3255

Course: Advanced Computer Network

Motivation

Design

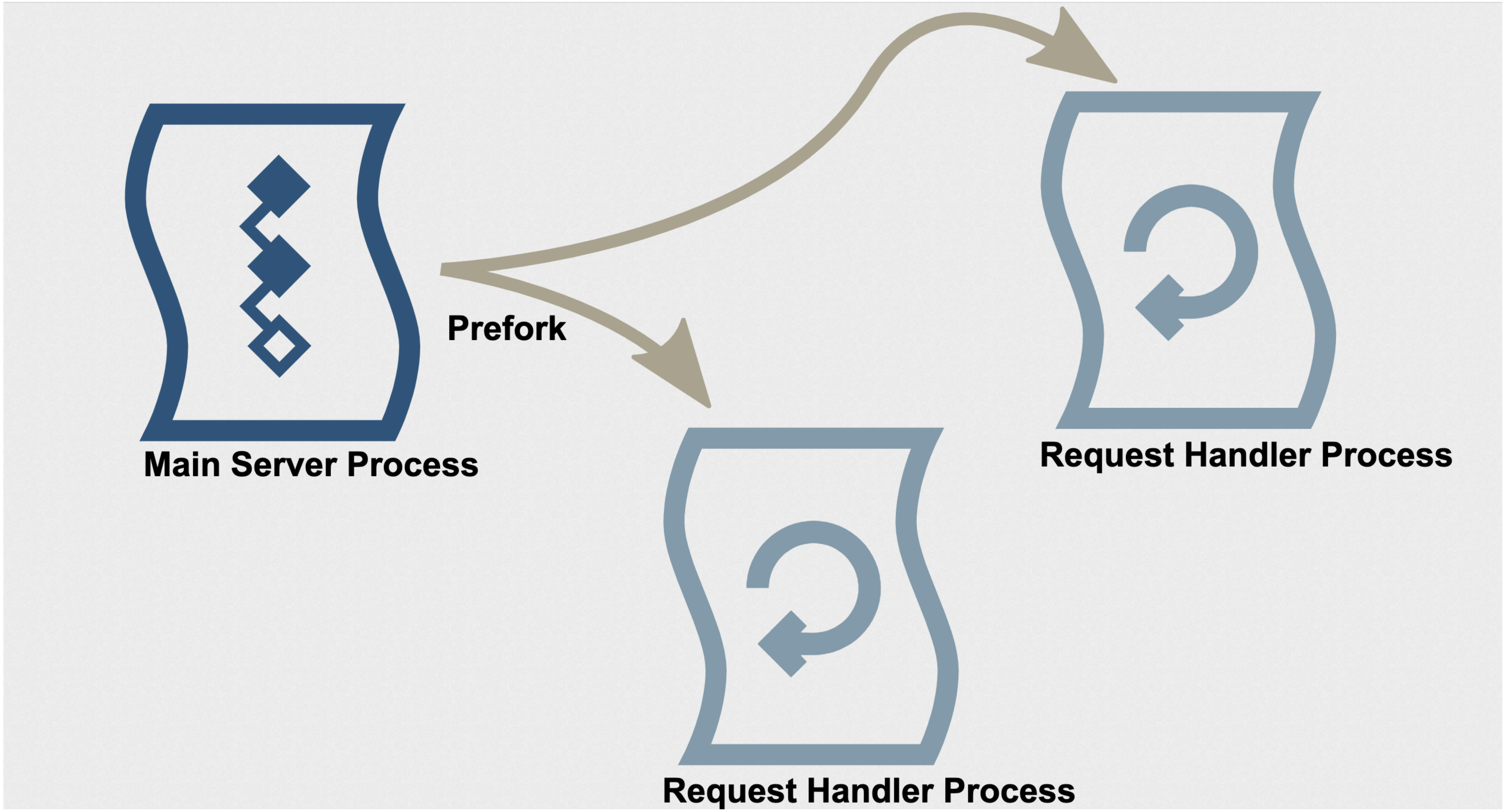
Implementation

Application Demo Show

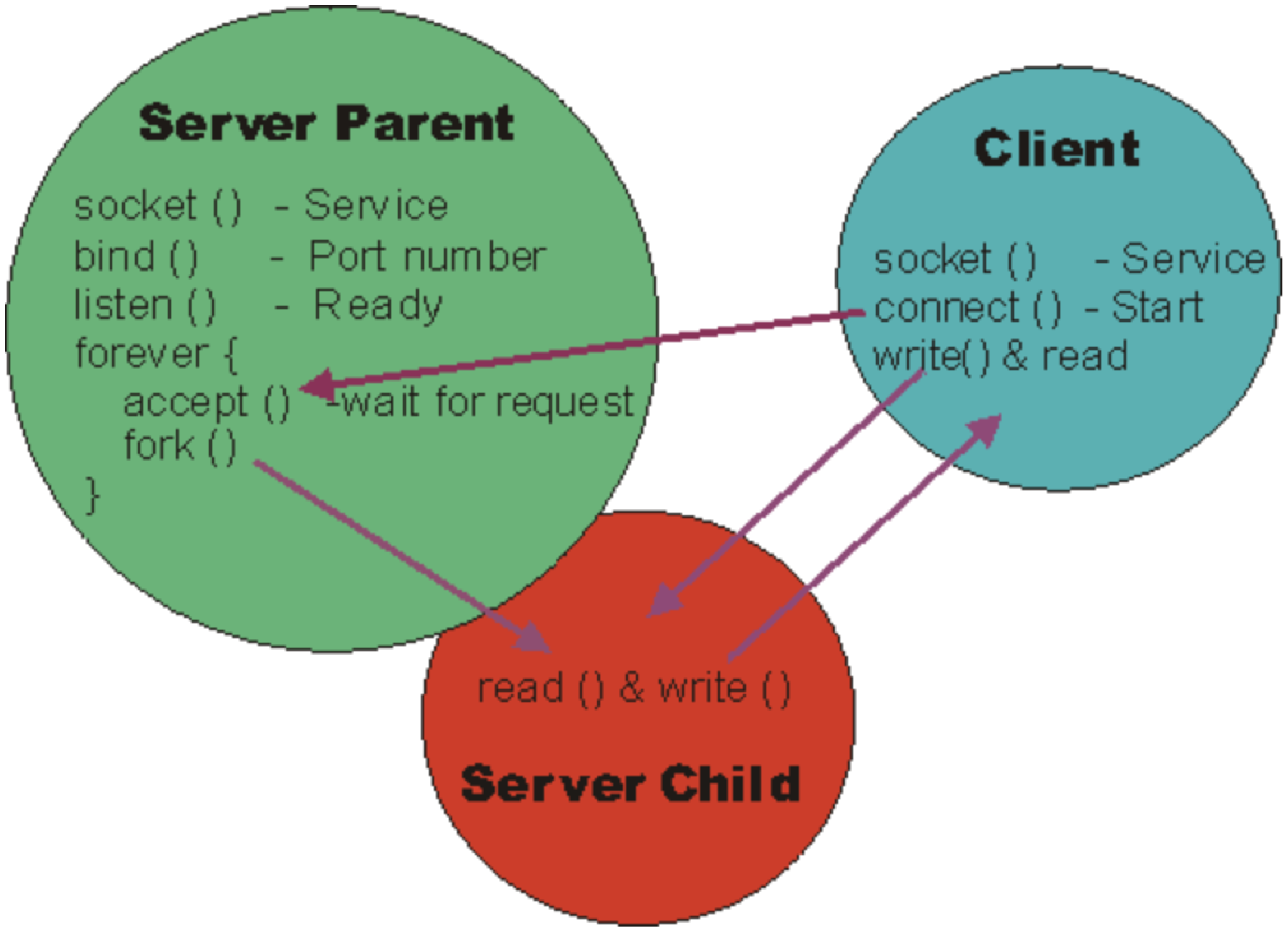
Motivation

1. Have a deep understanding of how a web server works.
2. Dive into HTTP protocol.
3. Get familiar with socket programming under Linux.
4. Get familair with C for system programming under Linux.

Design: Multi-Threading Based Model



Design: Socket Creation, Listening and Connection



Implementation: socket and pthread interface

- socket()
- bind()
- accept()
- connect()
- send()
- recv()
- pthread_create()

Implementation: Data Structure

```
typedef struct Header {
    char *name;
    char *value;
    struct Header *next;
} Header;

typedef struct Request {
    char method[128];
    char url[128];
    char version[128];
    struct Header *headers;
    char *body;
} Request;
```

Implementation: startServer

```
int startServer(u_short servPort) {

    //1. create a server socket
    servSocket = socket(PF_INET, SOCK_STREAM, 0);


    struct sockaddr_in servAddr;

    //2. bind the current port to the server socket
    if (bind(servSocket, (struct sockaddr *)&servAddr, sizeof(servAddr)) < 0) {
        error("bind socket fails");
    }

    //3. start listening  requests.
    if (listen(servSocket, 30) < 0) {
        error("listen socket fails");
    }

    return servSocket;
}
```

Implementation: serve

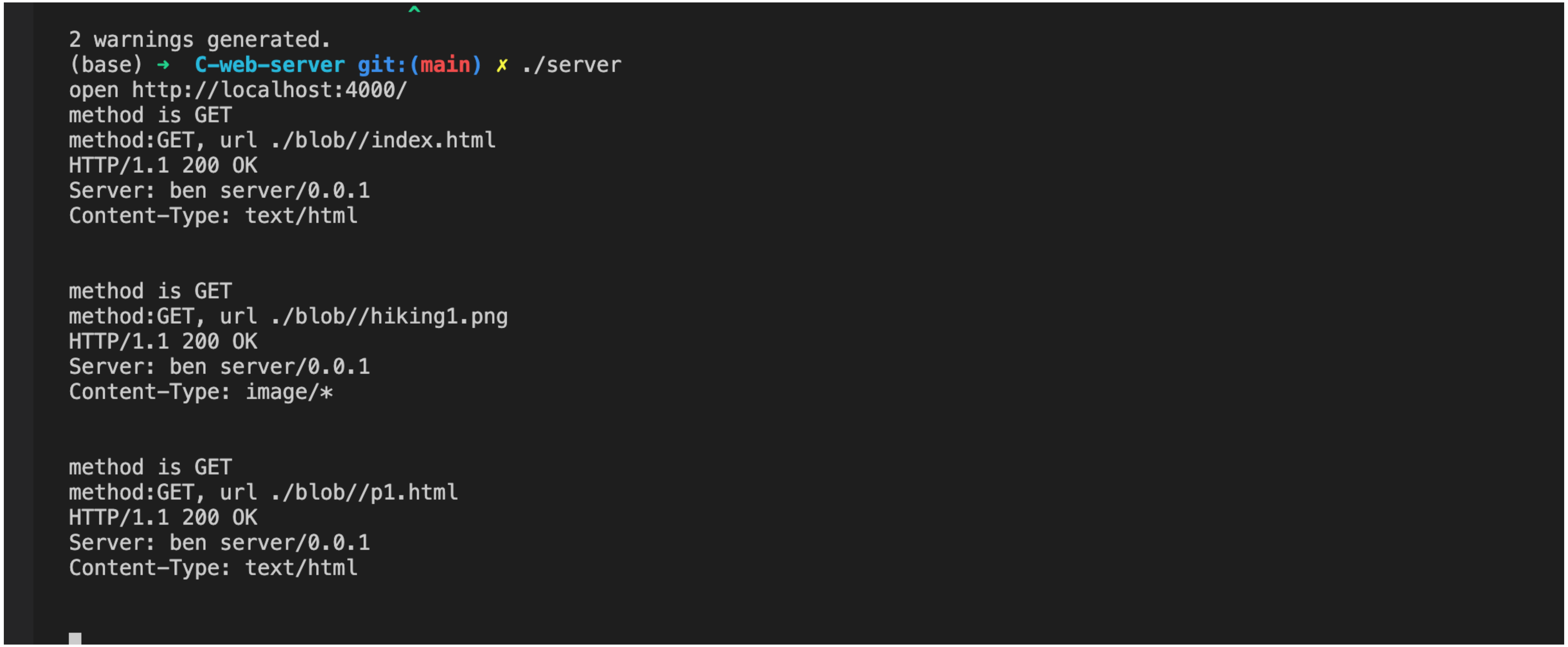
```
void serve(int servSocket) {
    while(1) {

        //1. accept client socket
        int clientSocket = accept(servSocket, (struct sockaddr *) &clientAddr, &clientAddrLen);
            ...
            ...
        //2. fork a child thread and then handle request.
        int createReuslt = pthread_create(&thread, NULL, (void *)requestHandler, (void *)(intptr_t)clientSocket);
        if (createReuslt != 0) {
            error("pthread creation fails");
        }
    }

    close(servSocket);
}
```

Implementation: requestHandler

```c
void requestHandler(void *arg)
{
    int client = (intptr_t)arg;
    char buf[1024];

    //1. parse a request
    Request* request = parseRequest(client, buf);

    printf("method:%s, url %s\n",request->method,  request->url);

    ```
    ```
    ```
    //2. get the static files
    sendStaticFiles(client, request->url, request->url);

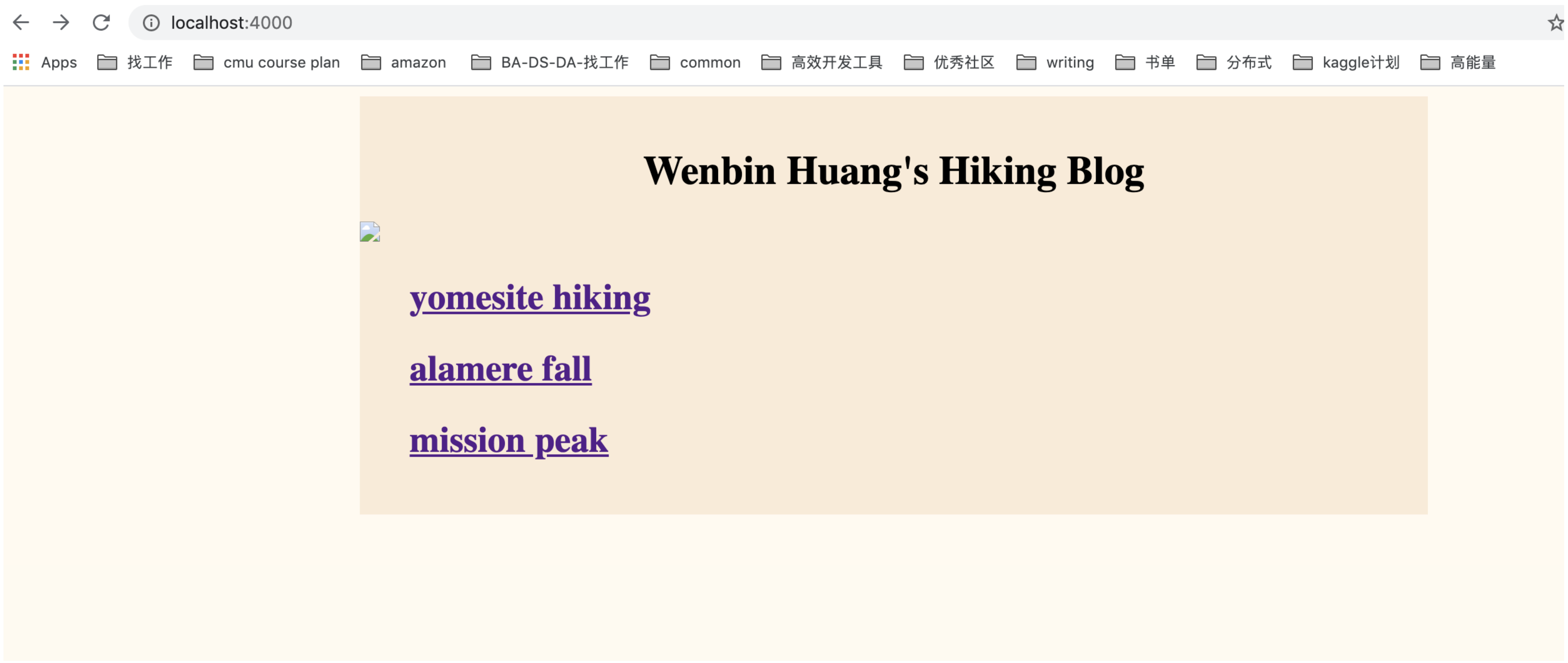
    free(request);
    close(client);
}
```
```

Application Demo Show: start web server



```
2 warnings generated.
(base) ➜  C-web-server git:(main) ✗ ./server
open http://localhost:4000/
method is GET
method:GET, url ./blob//index.html
HTTP/1.1 200 OK
Server: ben server/0.0.1
Content-Type: text/html


method is GET
method:GET, url ./blob//hiking1.png
HTTP/1.1 200 OK
Server: ben server/0.0.1
Content-Type: image/*


method is GET
method:GET, url ./blob//p1.html
HTTP/1.1 200 OK
Server: ben server/0.0.1
Content-Type: text/html
```

Application Demo Show: A Simple Hiking Blog



localhost:4000

Apps 找工作 cmu course plan amazon BA-DS-DA-找工作 common 高效开发工具 优秀社区 writing 书单 分布式 kaggle计划 高能量

Wenbin Huang's Hiking Blog

yomesite hiking

alamere fall

mission peak

Thanks

Reference

1. https://berb.github.io/diploma-thesis/original/042_serverarch.html
2. https://www.ibm.com/developerworks/systems/library/es-nweb/index.html