CSCI315 – Operating Systems Design

Department of Computer Science

Bucknell University

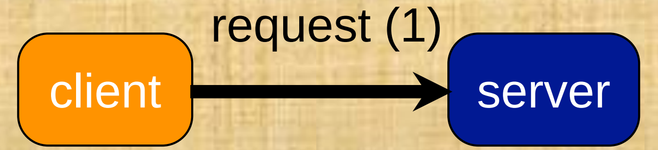
Examples of Thread and Process

This set of notes is based on notes from the textbook authors, as well as L. Felipe Perrone, Joshua Stough, and other instructors.
Xiannong Meng, Fall 2021.

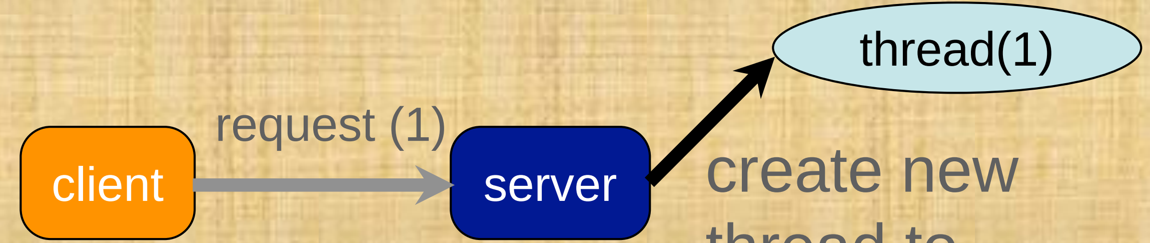
Examples of Thread and Process

- There are many advantages (or motivations) to use the thread architecture.
- One of them is to have a server process, in which each thread can serve one task.
- We’ll examine a couple such examples in this section.

Multithreaded Server Architecture



Multithreaded Server Architecture



client

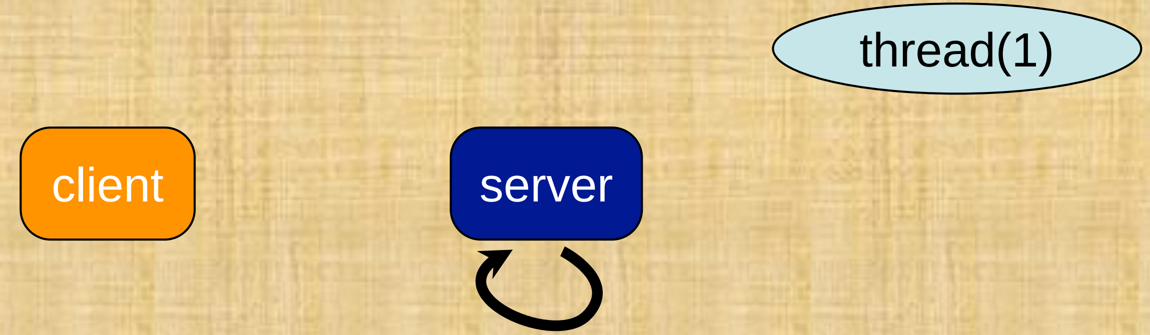
request (1)

server

thread(1)

create new thread to service request

Multithreaded Server Architecture



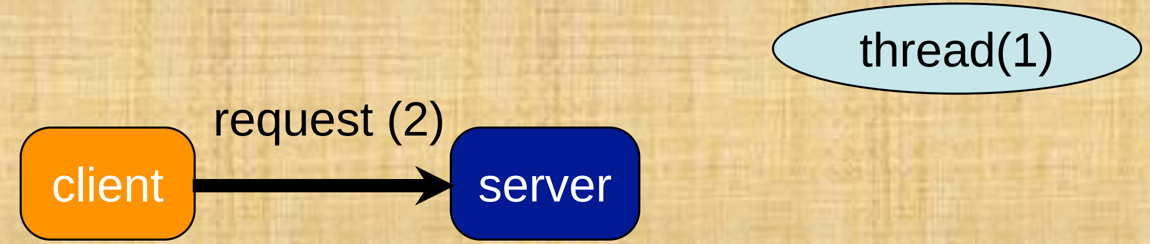
thread(1)

client

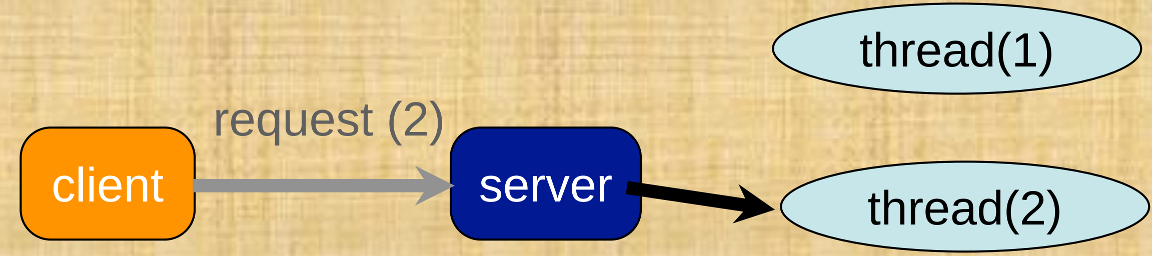
server

resume
listening for
new requests

Multithreaded Server Architecture

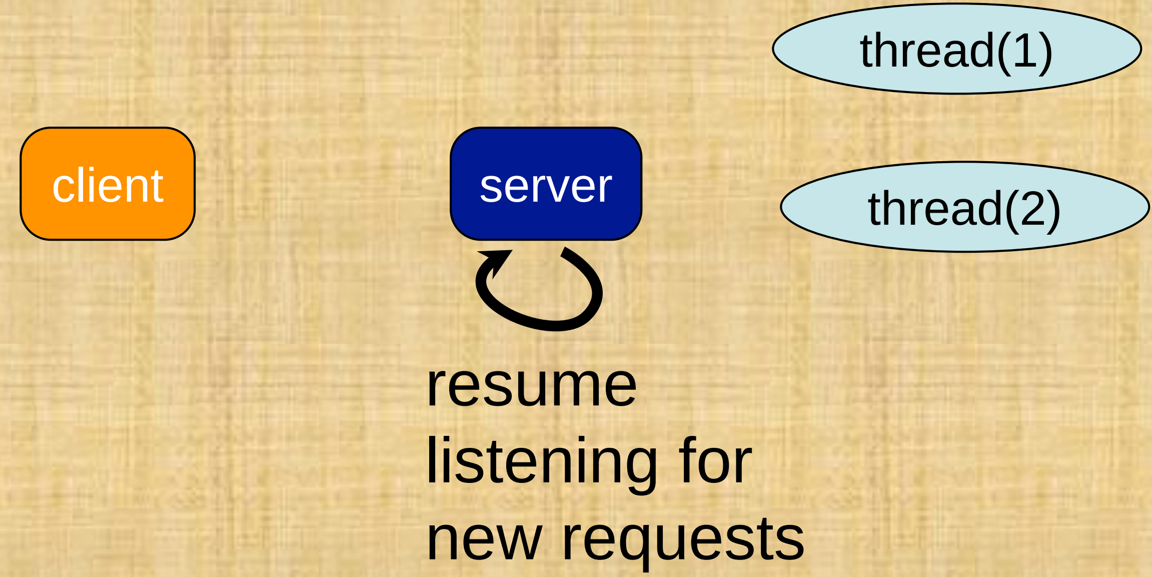


Multithreaded Server Architecture



create new
thread to
service request

Multithreaded Server Architecture

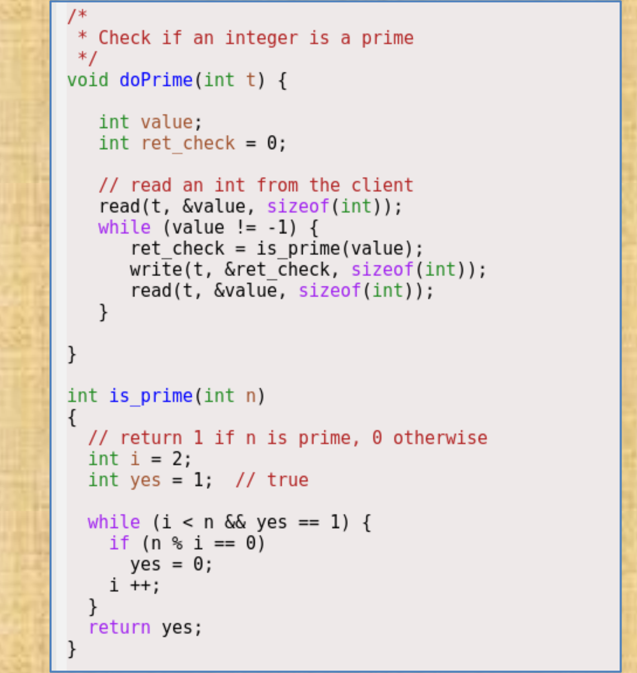


Each of these threads can serve a request. These requests can be of the same type, or they can be different.

Multithread Server Example

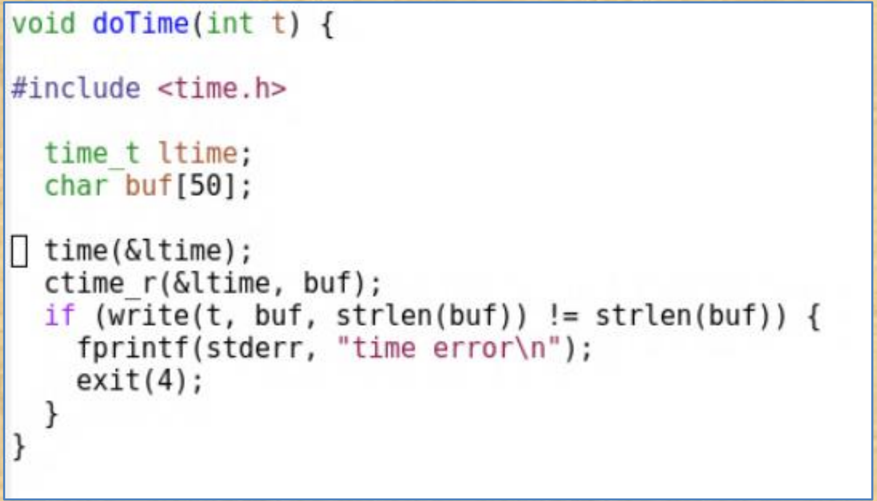
- A server is a program that can provide some kind of services to clients.
- Assume we have a server that can provide two simple services, **time** and **is_prime**.
- The **time** service sends the local time to the client.
- The **is_prime** service accepts an integer from the client and returns a value of 1 if the number is a prime, 0 otherwise.

Services Provided



```c
/*
 * Check if an integer is a prime
 */
void doPrime(int t) {

   int value;
   int ret_check = 0;

   // read an int from the client
   read(t, &value, sizeof(int));
   while (value != -1) {
      ret_check = is_prime(value);
      write(t, &ret_check, sizeof(int));
      read(t, &value, sizeof(int));
   }

}

int is_prime(int n)
{
  // return 1 if n is prime, 0 otherwise
  int i = 2;
  int yes = 1;  // true

  while (i < n && yes == 1) {
    if (n % i == 0)
      yes = 0;
    i ++;
  }
  return yes;
}
```

Key part of the **is_prime** service



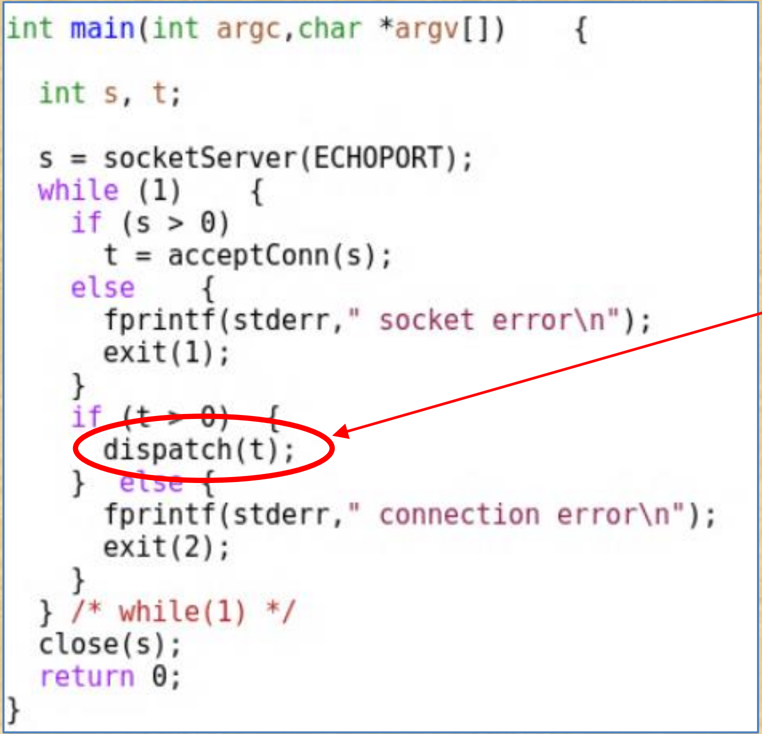
```c
void doTime(int t) {

#include <time.h>

  time_t ltime;
  char buf[50];

□ time(&ltime);
  ctime_r(&ltime, buf);
  if (write(t, buf, strlen(buf)) != strlen(buf)) {
    fprintf(stderr, "time error\n");
    exit(4);
  }
}
```

The complete **time** service

http://www.eg.bucknell.edu/~cs315/F2021/meng/code/time-prime/sevices.c

http://www.eg.bucknell.edu/~cs315/F2021/meng/code/time-prime/

A Single Thread Server



```
int main(int argc,char *argv[])    {

  int s, t;

  s = socketServer(ECHOPORT);
  while (1)    {
    if (s > 0)
      t = acceptConn(s);
    else     {
      fprintf(stderr," socket error\n");
      exit(1);
    }
    if (t > 0)   {
      dispatch(t);
    }  else {
      fprintf(stderr," connection error\n");
      exit(2);
    }
  } /* while(1) */
  close(s);
  return 0;
}
```

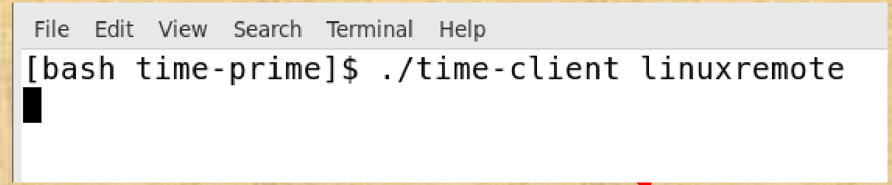
One service at a time, the next service will have to wait until the previous one completes.

```
[bash time-prime]$ make time-prime-server
gcc     -g -Wall -Werror     -c time-prime-server.c
gcc     -g -Wall -Werror     -c tcplib.c
gcc     -g -Wall -Werror     -c services.c
gcc     -o time-prime-server time-prime-server.o tcplib.o services.o -lpthread
[bash time-prime]$ ./time-prime-server &
[2] 6432
[bash time-prime]$
```

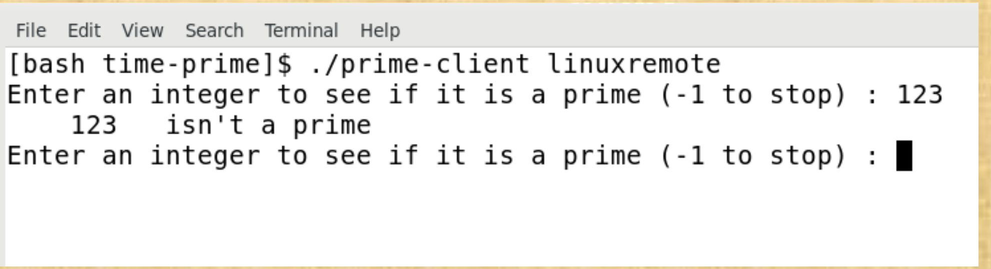
compile and execute **time-primeserver** first

http://www.eg.bucknell.edu/~cs315/F2021/meng/code/time-prime/time-prime-server.c

Single Service Screenshot



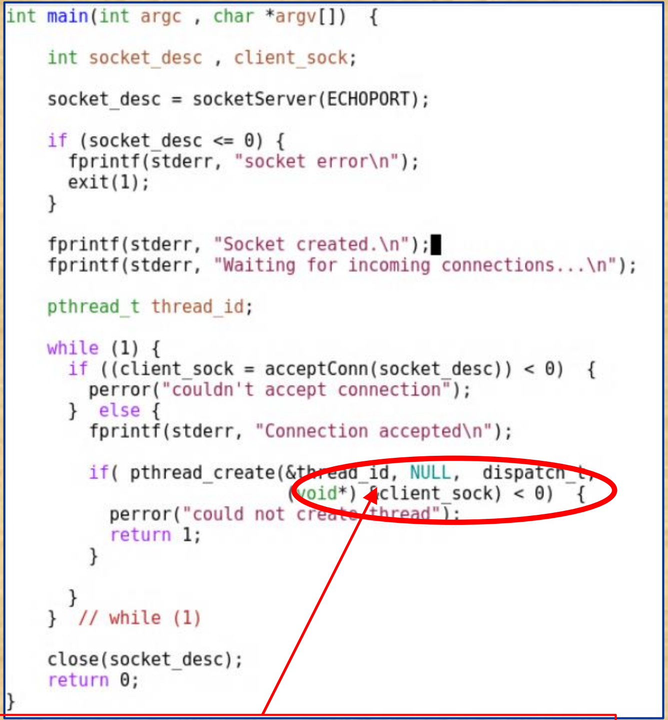
```
[bash time-prime]$ ./time-client linuxremote
█
```



```
[bash time-prime]$ ./prime-client linuxremote
Enter an integer to see if it is a prime (-1 to stop) : 123
    123   isn't a prime
Enter an integer to see if it is a prime (-1 to stop) : █
```

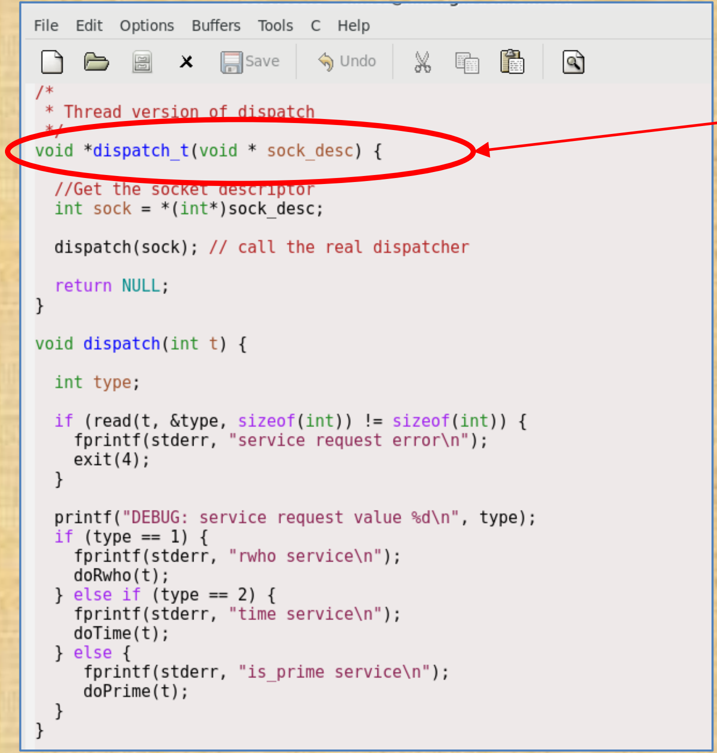
We requested “prime” first. The “time” request will not get serviced until the “prime” service is completed.

Multithread Server



```
int main(int argc , char *argv[])  {

    int socket_desc , client_sock;

    socket_desc = socketServer(ECHOPORT);

    if (socket_desc <= 0) {
      fprintf(stderr, "socket error\n");
      exit(1);
    }

    fprintf(stderr, "Socket created.\n");
    fprintf(stderr, "Waiting for incoming connections...\n");

    pthread_t thread_id;

    while (1) {
      if ((client_sock = acceptConn(socket_desc)) < 0)  {
        perror("couldn't accept connection");
      }  else {
        fprintf(stderr, "Connection accepted\n");

        if( pthread_create(&thread_id, NULL,  dispatch_t,
                                (void*) &client_sock) < 0)  {
          perror("could not create thread");
          return 1;
        }

      }
    }  // while (1)

    close(socket_desc);
    return 0;
}
```

Multi-threaded server with parameter

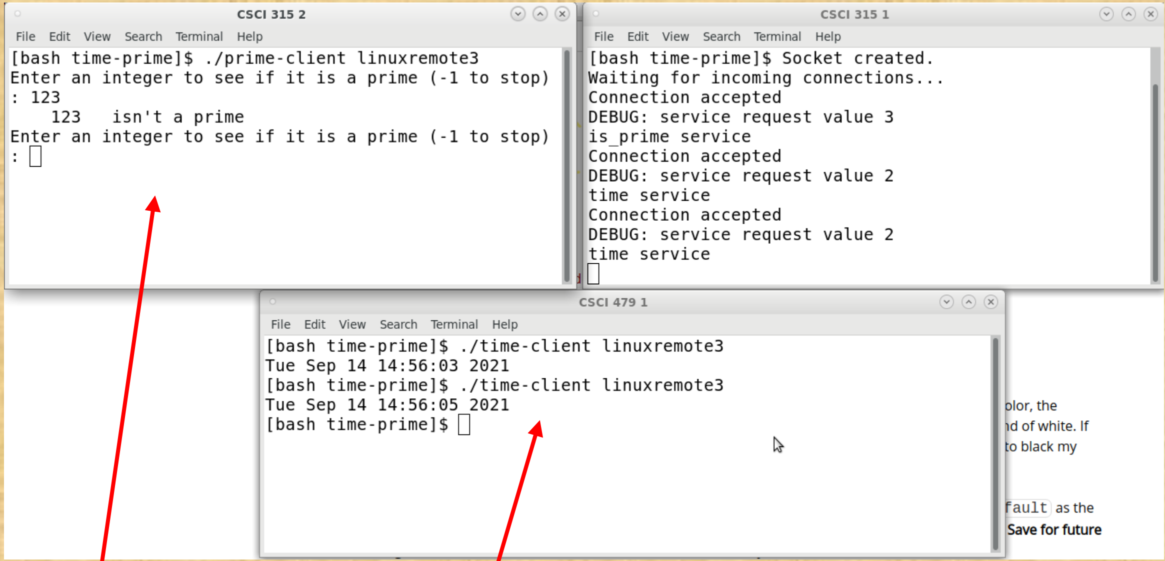


```
/*
 * Thread version of dispatch
 */
void *dispatch_t(void * sock_desc) {

  //Get the socket descriptor
  int sock = *(int*)sock_desc;

  dispatch(sock); // call the real dispatcher

  return NULL;
}

void dispatch(int t) {

  int type;

  if (read(t, &type, sizeof(int)) != sizeof(int)) {
    fprintf(stderr, "service request error\n");
    exit(4);
  }

  printf("DEBUG: service request value %d\n", type);
  if (type == 1) {
    fprintf(stderr, "rwho service\n");
    doRwho(t);
  } else if (type == 2) {
    fprintf(stderr, "time service\n");
    doTime(t);
  } else {
     fprintf(stderr, "is_prime service\n");
     doPrime(t);
  }
}
```

Threaded dispatcher determining service type

http://www.eg.bucknell.edu/~cs315/F2021/meng/code/time-prime/time-prime-server-thread.c

Multithread Server Screenshot



CSCI 315 2
File Edit View Search Terminal Help
CSCI 315 1
File Edit View Search Terminal Help
CSCI 479 1
File Edit View Search Terminal Help
olor, the
nd of white. If
to black my
fault as the
Save for future

```
[bash time-prime]$ ./prime-client linuxremote3
Enter an integer to see if it is a prime (-1 to stop)
: 123
    123   isn't a prime
Enter an integer to see if it is a prime (-1 to stop)
:
```

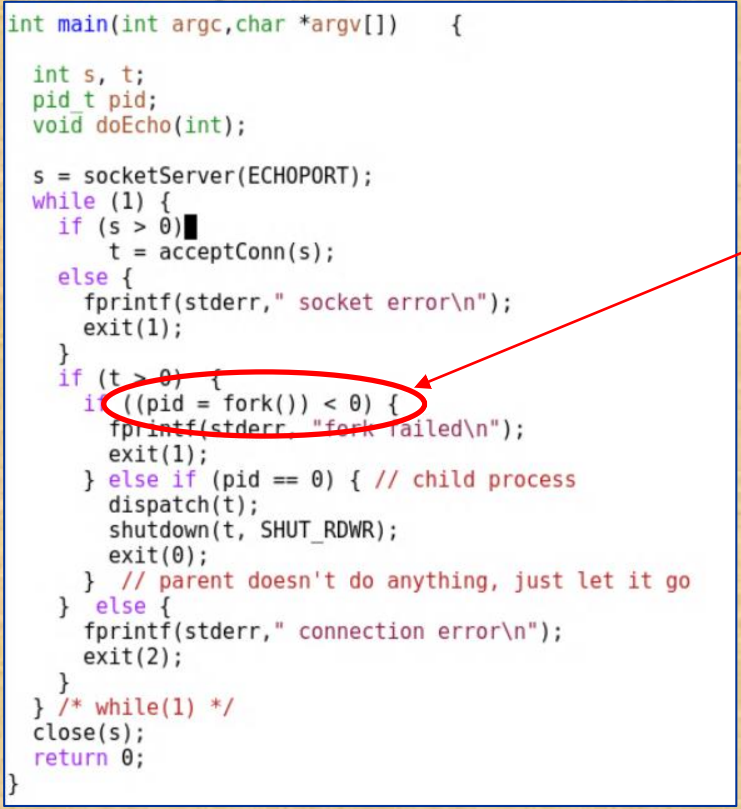
```
[bash time-prime]$ Socket created.
Waiting for incoming connections...
Connection accepted
DEBUG: service request value 3
is_prime service
Connection accepted
DEBUG: service request value 2
time service
Connection accepted
DEBUG: service request value 2
time service
```

```
[bash time-prime]$ ./time-client linuxremote3
Tue Sep 14 14:56:03 2021
[bash time-prime]$ ./time-client linuxremote3
Tue Sep 14 14:56:05 2021
[bash time-prime]$
```

We requested “is_prime” first. The “time” request can be serviced at the same time. Any number of these services can be requested and will be serviced simultaneously.

```
[bash time-prime]$ ./time-prime-server-thread &
[1] 20533
[bash time-prime]$ Socket created.
Waiting for incoming connections...
Connection accepted
DEBUG: service request value 3
is_prime service
……
[bash time-prime]$
```

Multi-process Server

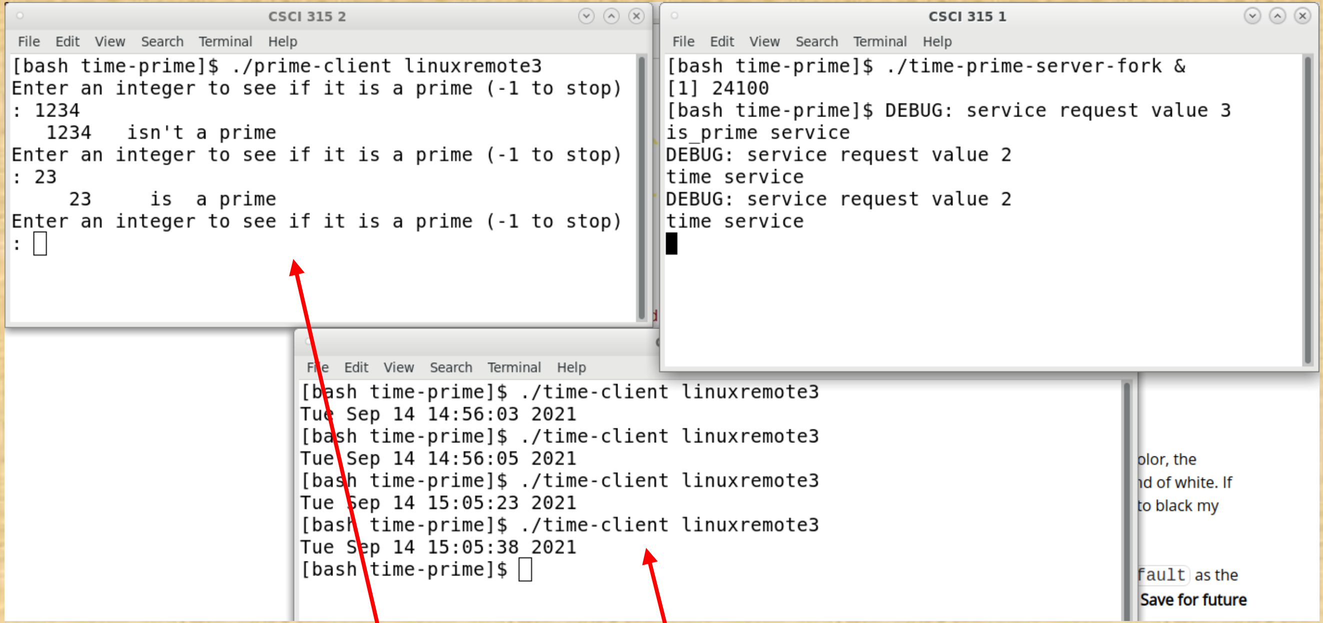


```c
int main(int argc,char *argv[])    {

  int s, t;
  pid_t pid;
  void doEcho(int);

  s = socketServer(ECHOPORT);
  while (1) {
    if (s > 0)
        t = acceptConn(s);
    else {
      fprintf(stderr," socket error\n");
      exit(1);
    }
    if (t > 0)   {
      if ((pid = fork()) < 0) {
        fprintf(stderr, "fork failed\n");
        exit(1);
      } else if (pid == 0) { // child process
        dispatch(t);
        shutdown(t, SHUT_RDWR);
        exit(0);
      }  // parent doesn't do anything, just let it go
    }  else {
      fprintf(stderr," connection error\n");
      exit(2);
    }
  } /* while(1) */
  close(s);
  return 0;
}
```

We can use `fork()` to create one process for each request.

http://www.eg.bucknell.edu/~cs315/F2021/meng/code/time-prime/time-prime-server-fork.c

Multi-process Server Screenshot


CSCI 315 2
File Edit View Search Terminal Help
CSCI 315 1
File Edit View Search Terminal Help
File Edit View Search Terminal Help
olor, the
nd of white. If
to black my
fault as the
Save for future

```
[bash time-prime]$ ./prime-client linuxremote3
Enter an integer to see if it is a prime (-1 to stop)
: 1234
   1234   isn't a prime
Enter an integer to see if it is a prime (-1 to stop)
: 23
     23     is  a prime
Enter an integer to see if it is a prime (-1 to stop)
: 
```

```
[bash time-prime]$ ./time-prime-server-fork &
[1] 24100
[bash time-prime]$ DEBUG: service request value 3
is_prime service
DEBUG: service request value 2
time service
DEBUG: service request value 2
time service
```

```
[bash time-prime]$ ./time-client linuxremote3
Tue Sep 14 14:56:03 2021
[bash time-prime]$ ./time-client linuxremote3
Tue Sep 14 14:56:05 2021
[bash time-prime]$ ./time-client linuxremote3
Tue Sep 14 15:05:23 2021
[bash time-prime]$ ./time-client linuxremote3
Tue Sep 14 15:05:38 2021
[bash time-prime]$ 
```

We requested “is_prime” first. The “time” request can be serviced at the same time. Any number of these services can be requested and will be serviced simultaneously. The effect is similar to multi-threaded service, except now processes are used.