Lab-5:Concurrent Server

Dr. Asif Uddin Khan

Concurrent Server

- A **concurrent server** handles multiple clients at the same time.
- The **fork** system call is invoked for creating one child process for each client.
- Concurrent server should run continuously to process client requests

How to create concurrent server

- Keep accept function and other codes within infinite loop in server side code.
- Create child process for each client using fork() function

Example:

```
while (1) {
 newsockfd = accept(sockfd, (struct sockaddr *) &cli_addr, &clilen);
//Other codes
pid = fork();
//Other codes
}
```

Concurrent Server

```
while (1) {
    newsockfd = accept(sockfd, (struct sockaddr *) &cli_addr,(socklen_t*) &clilen);
    if (newsockfd < 0) {
        perror("ERROR on accept");
        exit(1);
    }
    /* Create child process */
    pid = fork();

    if (pid < 0) {
        perror("ERROR on fork");
        exit(1);
    }
    if (pid == 0) {
        /* This is the client process */
        //close(sockfd);
        doprocessing(newsockfd);
        exit(0);
    }
    else {
        close(newsockfd);
    }
} /* end of while */
}
```

Concurrent Server

```
void doprocessing (int sock) {
    int n;
    char buffer[256];
    bzero(buffer,256);
    n = read(sock,buffer,255);
    if (n < 0) {
        perror("ERROR reading from socket");
        exit(1);
    }
    printf("Here is the message: %s\n",buffer);
    n = write(sock,"I got your message",18);

    if (n < 0) {
        perror("ERROR writing to socket");
        exit(1);
    }
}
```

Calculator Server

Dr. Asif Uddin Khan

Take integer value from keyboard and send using socket

```
scanf("%d",&num); // Read integer value from keyboard
write(sockfd,&num,sizeof(int));//Attach the integer value
                               //“num” to the socket
                               //pointed by “sockfd”
```

Read value from socket

- read(sockfd, &num, sizeof(int)); // Read from socket
  //pointed by “sockfd” and
  //write to the variable
  //“num”
- printf("The result is %d \n",num); // display number the
  //value of “num”

Client

```
printf("Enter first number:");
scanf("%d",&num1);
write(sockfd,&num1,sizeof(int));
printf("Enter second number:");
scanf("%d",&num2);
write(sockfd,&num2,sizeof(int));
printf("Enter the operator:Add->1 Sub->2 Mul->3 Div->4:");
scanf("%d",&op);
write(sockfd,&op,sizeof(int));
//Read result from server
read(sockfd,&result,sizeof(int));
printf("The result is %d \n",result);
```

Server

```
read(sock,&num1,sizeof(int));
printf("first number received: %d\n",num1);
read(sock,&num2,sizeof(int));
printf("second number received: %d\n",num2);
read(sock,&op,sizeof(int));
  if(op==1){
      printf("Operator received: Add");
      result=num1+num2;}
  if(op==2){
      printf("Operator received: Sub");
      result=num1-num2;}
  if(op==3){
      printf("Operator received: Mul");
      result=num1*num2;}
  if(op==4){
      printf("Operator received: Div");
      result=num1*num2;}
  write(sock,&result,sizeof(int));
```

Assignment-4.1

- Design a TCP client and server program, where the server program runs always. Server receives messages from continuously from multiple clients and displays it without stopping the client.

Assignment 4.2

- Write a menu driven TCP socket program for following task.
- Client should read 2 integer numbers and one operator from keyboard and send to the server for calculation. After receiving the numbers and operator the server should perform the operation and send the result back to client. Then client should display the result.

Note: You can modify concurrent server program

References

- https://www.tutorialspoint.com/unix_sockets/index.htm
- http://home.iitk.ac.in/~chebrolu/ee673-f06/sockets.pdf