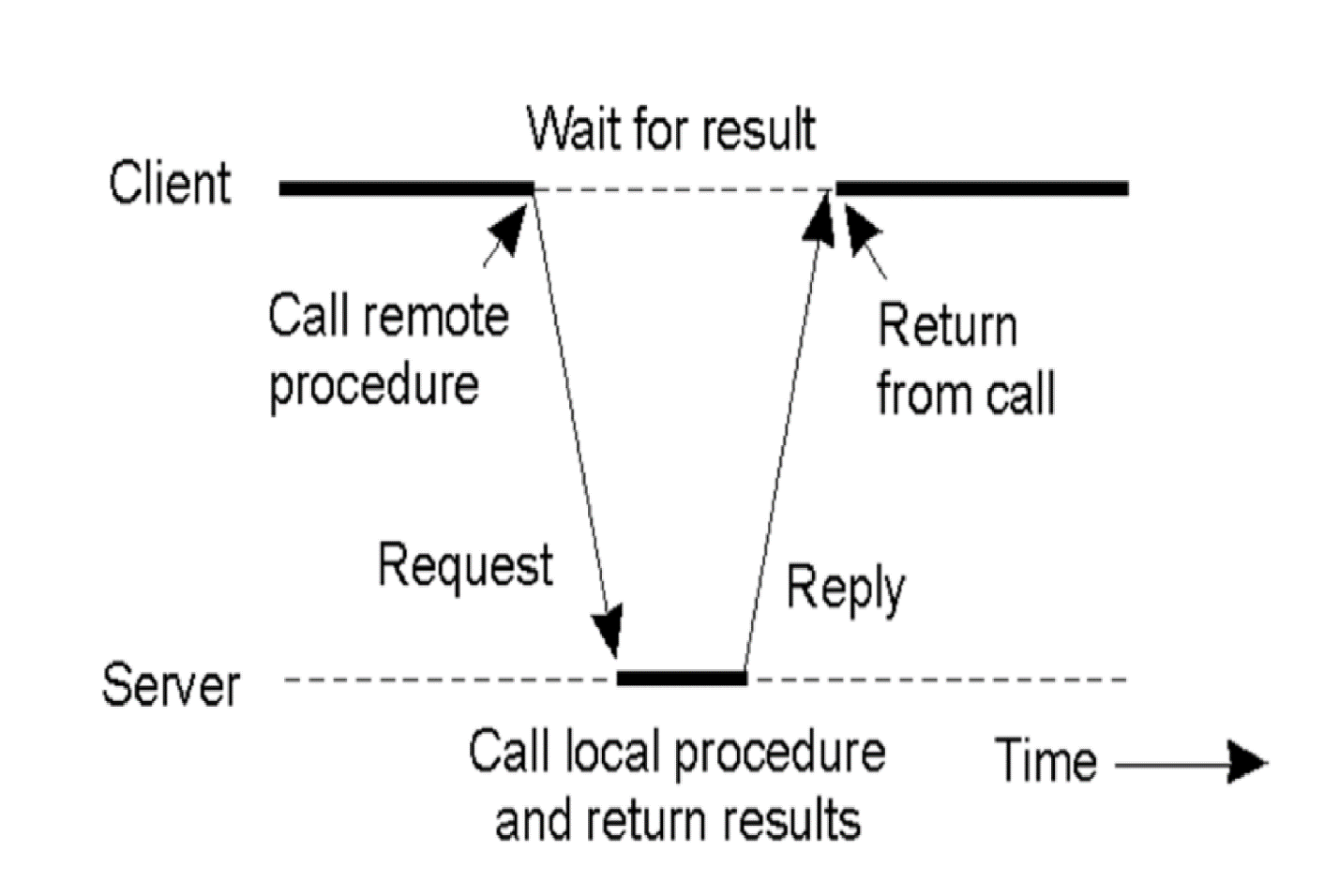
**System design document for Project\_1 of CprE 550**

In this programming assignment we are going to implement a network management application that tracks user logins, CPU usage and other statistics on a host and allows querying by a RPC-based network management system. We will implement the synchronous version of the RPC for our project and track Current system time, CPU usage, Memory usage and Load procs per min of the server.

**RPC Synchronous Call Diagram**



**Passing Value parameters**

Server process

Client process

6.Stub makes local call to date

1.Client call to procedure with user choice in option variable

K=date(\*option)

K=date(\*option)

5.Stub unpacks message

2.Stub builds message

Proc=”date”

Proc=”date”

4.Sever OS hands message to server stub

int : option

int : option

3.Message is sent across network

Server OS

Client OS

The above steps indicate how current system time is obtained from the server system by the client calling the server-side date function which gives you the current date, time and day. The above steps are repeated for different procedures like-

1. void memory(CLIENT \*clnt) : This function calls the server side memory\_1(void) function which obtains details like page size, number of physical pages, available amount of physical pages, total swap size, available swap size, total ram, shared memory and buffer memory in the system.
2. void cpu(CLIENT \*clnt) : This function calls the server side cpu\_1(void) function which obtains details like one minute , five minute and fifteen minute load averages of the server system.
3. void process(CLIENT \*clnt) : This function calls the server side process\_1(void) function which obtains details like system boot time and number of processes of the server system.

**Files and their description-**

1.date.x- specification for remote date, time, date and time, memory, cpu and process service

2.date\_clnt.c - client-side stub

3.date\_svc.c – server-side

4.client.c - client program that invokes the server-side functions

5. server.c - server program that implements the functions that get information about system time, CPU usage and memory usage

6. date.h -header file for constants, data types and definitions of remote procedures.

7. client and server – executable files

**Function description-**

1. void date(\* option, clnt)

The function passes the address of the variable holding the choice specified by the user which selects date, time and date and time of the server.

clnt is the client handle that is passed to call the appropriate server function.

This function calls the server side date\_1(\* option) function which obtains the current system date, time or date and time as per the choice of the user which is passed in the option parameter and server side function copies the system time information into a character buffer and then returns a pointer to this buffer in a network message to the client calling function where the date, time or date and time is displayed.

1. void memory(NULL, clnt)

clnt is the client handle that is passed to call the appropriate server function.

This function calls the server side memory\_1(void) function which obtains information about the server like number of physical pages in the memory, page size, the total memory in use, total number of available physical pages in the memory, buffer memory , shared memory and the total free memory by calling sysinfo(struct sysinfo \*info) and functions like get\_phys\_pages(void), getpagesize(void), get\_avphys\_pages(void) and then returns a pointer to a double to the calling client side function in a message over the network.

1. void process(NULL, clnt)

clnt is the client handle that is passed to call the appropriate server function.

This function calls the server side process\_1(void) function which obtains information about the server like one minute load average, five minute load average and fifteen minute load average of the system by calling sysinfo(struct sysinfo \*info) and number of elements written to loadavgs by using getloadavg(double loadavg[], int nelem) and then returns a pointer to a double to the calling client side function in a message over the network.

1. void cpu(NULL, clnt)

clnt is the client handle that is passed to call the appropriate server function.

This function calls the server side cpu\_1(void) function which obtains information about the server like time in seconds since boot of the system, number of processes running in the system and then returns a pointer to a double to the calling client-side function in a message over the network.