

Name: Brijesh Rameshbhai Rohit

Admission number: U19CS009

## DISTRIBUTED SYSTEMS

### ASSIGNMENT - 06

Simulate RPC (Create any one procedure on remote machine and call it from local machine)

List of programs for RPC

1. Find out the factorial of given number.

CODE :

```
struct number
{
    int num;
};

program FACTORIAL_PROG{
    version FACTORIAL_VERS{
        int factorial(number)=1;
    }=1;
}=0x84562877;
```

CODE - SERVER

```
/*
 * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.
 */

#include "factorial.h"

int *
factorial_1_svc(number *argp, struct svc_req *rqstp)
{
    static int result;

    /*
     * insert server code here
     */
}
```

```

    */

    int flag = 1;
    if(argp->num < 0)
    {
        printf("Factorial of negative numbers are not calculated!!!\n");
        flag = 0;
    }
    result = 1;
    for(int i = 2 ; i <= argp->num ; i++)
        result *= i;
    if(flag)
        printf("%d! = %d\n", argp->num, result);
    return &result;
}

```

#### CODE - CLIENT

```

/*
 * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.
 */

#include "factorial.h"

void
factorial_prog_1(char *host, int num1)
{
    CLIENT *clnt;
    int *result_1;
    number factorial_1_arg;

#ifdef DEBUG
    clnt = clnt_create (host, FACTORIAL_PROG, FACTORIAL_VERS, "udp");
    if (clnt == NULL) {
        clnt_pcreateerror (host);

```

```

        exit (1);
    }
#endif /* DEBUG */

    int flag = 1;
    factorial_1_arg.num = num1;
    result_1 = factorial_1(&factorial_1_arg, clnt);
    if (result_1 == (int *) NULL) {
        clnt_perror (clnt, "call failed");
    }
    else
    {
        if(num1 < 0)
        {
            printf("Factorial of negative is not possible!!!\n");
            flag = 0;
        }
        else
        {
            printf("Factorial of %d : %d\n", num1, *result_1);
        }
    }
#endif /* DEBUG */
    clnt_destroy (clnt);
#endif /* DEBUG */
}

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

    if (argc < 2) {
        printf ("usage: %s server_host\n", argv[0]);
        exit (1);
    }
    host = argv[1];
    int num1;

```

```

char ch = 'y';
while (ch=='y')
{
    system("clear");
    printf("Please enter an integer : ");
    scanf("%d", &num1);
    factorial_prog_1 (host, num1);
    printf("Do you want to calculate factorial again ? (y/n) : ");
    scanf("%s", &ch);
}
exit (0);
}

```

**OUTPUT :**

```

1: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/factorial ▾
brijesh@brijesh-VirtualBox:~/Documents/ds-assign06/factorial$ ./factorial_server
12! = 479001600
13! = 1932053504
█

2: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/factorial ▾
Please enter an integer : 13
Factorial of 13 : 1932053504
Do you want to calculate factorial again ? (y/n) : █

```

```

1: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/factorial ▾
brijesh@brijesh-VirtualBox:~/Documents/ds-assign06/factorial$ ./factorial_server
12! = 479001600
13! = 1932053504
-1! = 1
█

2: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/factorial ▾
Please enter an integer : -1
Factorial of negative is not possible!!!
Do you want to calculate factorial again ? (y/n) : n
brijesh@brijesh-VirtualBox:~/Documents/ds-assign06/factorial$ █

```

## 2. Implement Calculator (Basic operation).

CODE :

```
struct inputs{
    float num1;
    float num2;
    char operator;
};

program CALCULATE_PROG{
    version CALCULATE_VER{
        float CALCULATE(inputs)=1;
    }=1;
}=0x2fffffff;
```

CODE - SERVER

```
/*
 * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.
 */

#include "calculate.h"
float * calculate_1_svc(inputs *argp, struct svc_req *rqstp)
{
    static float result;
    float a = argp->num1;
    float b = argp->num2;
    char op = argp->operator;
    int flag = 1;
    if(op=='+')
        result = a+b;
    else if(op=='-')
        result = a-b;
    else if(op=='*')
        result = a*b;
    else
    {
        if(b==0)
        {
```

```

        printf("Division with 0 is invalid!!!\n");
        flag = 0;
    }
}

if(flag)
printf("%f %c %f = %f\n",a, op, b, result);

return &result;
}

```

#### CODE - CLIENT

```

/*
 * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.
 */

#include "calculate.h"

void
calculate_prog_1(char *host, float a, float b, char op)
{
    CLIENT *clnt;
    float *result_1;
    inputs calculate_1_arg;

#ifdef DEBUG
    clnt = clnt_create (host, CALCULATE_PROG, CALCULATE_VER, "udp");
    if (clnt == NULL) {
        clnt_pcreateerror (host);
        exit (1);
    }
#endif /* DEBUG */

    calculate_1_arg.num1 = a;

```

```

    calculate_1_arg.num2 = b;
    calculate_1_arg.operator = op;

    result_1 = calculate_1(&calculate_1_arg, clnt);
    if (result_1 == (float *) NULL) {
        clnt_perror (clnt, "call failed");
    }
    else
    {
        int flag = 1;
        if(op=='/')
        {
            if(b==0)
            {
                printf("Division with 0 is invalid!!!\n");
                flag = 0;
            }
        }
        if(flag)
        {
            printf("%f %c %f = %f",a, op, b, *result_1);
        }
    }
#ifdef DEBUG
    clnt_destroy (clnt);
#endif /* DEBUG */
}

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

    if (argc < 2) {
        printf ("usage: %s server_host\n", argv[0]);
        exit (1);
    }
    host = argv[1];

```

```
char op;
float a, b;
int ch=1;
while(ch)
{
    system("clear");
    printf("'+' : Addition\n");
    printf("'-' : Subtraction\n");
    printf("'*' : Multiplication\n");
    printf("'/' : Division\n");
    printf("Enter first number  : ");
    scanf("%f", &a);
    printf("Enter second number : ");
    scanf("%f", &b);
    printf("Enter operator : ");
    scanf("%s", &op);
    calculate_prog_1 (host, a, b, op);
    printf("\nDo calculations again ? (1/0) : ");
    scanf("%d", &ch);
}
printf("Hope you enjoyed our SERVICE!!\n");
exit (0);
}
```



OUTPUT :

```
brijesh@brijesh-VirtualBox:~/Documents/ds-assign06/calculator$ ./calculate_server
12.000000 + 13.000000 = 25.000000
```

```
2: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/calculator ▾
```

```
'+' : Addition
'-' : Subtraction
'*' : Multiplication
'/' : Division
Enter first number : 12
Enter second number : 13
Enter operator : +
12.000000 + 13.000000 = 25.000000
Do calculations again ? (1/0) : 
```

```
1: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/calculator ▾
```

```
brijesh@brijesh-VirtualBox:~/Documents/ds-assign06/calculator$ ./calculate_server
12.000000 + 13.000000 = 25.000000
1.123108 * 321.312317 = 360.868408
Division with 0 is invalid!!!
```

```
2: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/calculator ▾
```

```
'+' : Addition
'-' : Subtraction
'*' : Multiplication
'/' : Division
Enter first number : 1
Enter second number : 0
Enter operator : /
Division with 0 is invalid!!!

Do calculations again ? (1/0) : 0
Hope you enjoyed our SERVICE!!
```

```
1: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/calculator ▾
```

```
brijesh@brijesh-VirtualBox:~/Documents/ds-assign06/calculator$ ./calculate_server
12.000000 + 13.000000 = 25.000000
1.123108 * 321.312317 = 360.868408
```

```
2: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/calculator ▾
```

```
'+' : Addition
'-' : Subtraction
'*' : Multiplication
'/' : Division
Enter first number : 1.12311
Enter second number : 321.312321
Enter operator : *
1.123108 * 321.312317 = 360.868408
Do calculations again ? (1/0) : 
```

3. Find out whether given number is Prime Number or not.

CODE :

```
struct number{
    int num;
};

program PRIME_PROG{
    version PRIME_VERS{
        int prime(number)=1;
    }=1;
}=0x2f2f2f2f;
```

CODE - SERVER

```
/*
 * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.
 */

#include "prime.h"

int *
prime_1_svc(number *argp, struct svc_req *rqstp)
{
    static int result;
    result=1;
    if(argp->num < 1)
    {
        printf("Entered number is not positive!!\n");
        result = 0;
        return &result;
    }
    else if(argp->num == 1)
    {
        printf("1 is non-prime!!!\n");
        result = 0;
        return &result;
    }
}
```

```

    }
    int n = 1;
    while(n*n <= argp->num)
        n++;
    for(int i = 2 ; i <= n ; i++)
    {
        if(argp->num%i==0)
        {
            result = 0;
            printf("%d is non-prime!!!\n", argp->num);
            return &result;
        }
    }
    printf("%d is PRIME!!!\n", argp->num);
    return &result;
}

```

#### CODE - CLIENT

```

/*
 * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.
 */

#include "prime.h"

void
prime_prog_1(char *host, int num1)
{
    CLIENT *clnt;
    int *result_1;
    number prime_1_arg;

#ifdef DEBUG
    clnt = clnt_create (host, PRIME_PROG, PRIME_VERS, "udp");
    if (clnt == NULL) {

```

```

        clnt_pcreateerror (host);
        exit (1);
    }
#endif /* DEBUG */

    int flag = 1;
    prime_1_arg.num = num1;
    result_1 = prime_1(&prime_1_arg, clnt);
    if (result_1 == (int *) NULL) {
        clnt_perror (clnt, "call failed");
    }

    if(num1 < 1)
    {
        printf("Prime characteristic is only for Positive
integers!!!!\n");
    }
    else
    {
        if(*result_1)
        {
            printf("%d is a PRIME!!\n", num1);
        }
        else
        {
            printf("%d is non-prime\n", num1);
        }
    }

#ifdef DEBUG
    clnt_destroy (clnt);
#endif /* DEBUG */
}

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

```

```

    if (argc < 2) {
        printf ("usage: %s server_host\n", argv[0]);
        exit (1);
    }
    host = argv[1];
    char ch='y';
    int num;
    while(ch=='y')
    {
        system("clear");
        printf("Please enter an integer : ");
        scanf("%d", &num);
        prime_prog_1 (host, num);
        printf("Do you want to check more prime numbers ? (y/n) : ");
        scanf("%s", &ch);
    }

    exit (0);
}

```

**OUTPUT :**

```

1: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/prime ▾
brijesh@brijesh-VirtualBox:~/Documents/ds-assign06/prime$ ./prime_server
1323 is non-prime!!!
█

2: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/prime ▾
Please enter an integer : 1323
1323 is non-prime
Do you want to check more prime numbers ? (y/n) : █

```

```
1: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/prime ▾
brijesh@brijesh-VirtualBox:~/Documents/ds-assign06/prime$ ./prime_server
1323 is non-prime!!!
1234567 is non-prime!!!
█

2: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/prime ▾
Please enter an integer : 1234567
1234567 is non-prime
Do you want to check more prime numbers ? (y/n) : █
```

```
1: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/prime ▾
1323 is non-prime!!!
1234567 is non-prime!!!
13 is PRIME!!!
157 is PRIME!!!
█

2: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/prime ▾
Please enter an integer : 157
157 is a PRIME!!
Do you want to check more prime numbers ? (y/n) : n
brijesh@brijesh-VirtualBox:~/Documents/ds-assign06/prime$ █
```

```
1: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/prime ▾
brijesh@brijesh-VirtualBox:~/Documents/ds-assign06/prime$ ./prime_server
1323 is non-prime!!!
1234567 is non-prime!!!
13 is PRIME!!!
█

2: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/prime ▾
Please enter an integer : 13
13 is a PRIME!!
Do you want to check more prime numbers ? (y/n) : █
```

4. Print out the Fibonacci series till the given number.

CODE :

```
struct fibonacci{
    int n;
};

program FIB_PROG{
    version FIB_VERS{
        int fib(fibonacci)=1;
    }=1;
}=0x2f2f2f2f;
```

CODE - SERVER

```
/*
 * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.
 */

#include "fib.h"

int *
fib_1_svc(fibonacci *argp, struct svc_req *rqstp)
{
    static int result;

    int n=argp->n;
    if(n<=2)
        result=1;
    else
    {
        int cur=1,prev=1,i=2,temp;
        while(i<n)
        {
            temp=cur;
            cur=cur+prev;
            prev=temp;
            i++;
        }
    }
}
```

```

    }
    result=cur;
}
return &result;
}

```

## CODE - CLIENT

```

/*
 * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.
 */

#include "fib.h"

void
fib_prog_1(char *host, int n)
{
    CLIENT *clnt;
    int *result_1;
    fibonacci fib_1_arg;

#ifdef DEBUG
    clnt = clnt_create (host, FIB_PROG, FIB_VERS, "udp");
    if (clnt == NULL) {
        clnt_pcreateerror (host);
        exit (1);
    }
#endif /* DEBUG */

    fib_1_arg.n = 1;
    printf("Fibonacci series till %d : ", n);
    result_1 = fib_1(&fib_1_arg, clnt);
    if (result_1 == (int *) NULL) {
        clnt_perror (clnt, "call failed");
    }
}

```



```

else
{
    int i = 2;
    while (*result_1 < n)
    {
        printf(" %d", *result_1);
        fib_1_arg.n = i;
        result_1 = fib_1(&fib_1_arg, clnt);
        i++;
    }
    printf("\n");
}

#ifdef DEBUG
    clnt_destroy (clnt);
#endif /* DEBUG */
}

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

    if (argc < 2) {
        printf ("usage: %s server_host\n", argv[0]);
        exit (1);
    }
    host = argv[1];
    int n;
    char ch = 'y';
    while (ch == 'y')
    {
        system("clear");
        printf("Enter a number : ");
        scanf("%d", &n);
        fib_prog_1 (host, n);
        printf("Do you want to repeat? (y/n): ");
        scanf("%s", &ch);
    }
}

```

```
exit (0);  
}
```

OUTPUT :

```
1: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/fibonacci  
brijesh@brijesh-VirtualBox:~$ cd Documents/ds-assign06  
brijesh@brijesh-VirtualBox:~/Documents/ds-assign06$ cd fibonacci  
brijesh@brijesh-VirtualBox:~/Documents/ds-assign06/fibonacci$ ./fib_server  
Enter a number : 1000000000  
Fibonacci series till 1000000000 : 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597  
2584 4181 6765 10946 17711 28657 46368 75025 121393 196418 317811 514229 832040 1346269  
2178309 3524578 5702887 9227465 14930352 24157817 39088169 63245986  
Do you want to repeat? (y/n):  
  
1: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/fibonacci  
brijesh@brijesh-VirtualBox:~$ cd Documents/ds-assign06  
brijesh@brijesh-VirtualBox:~/Documents/ds-assign06$ cd fibonacci  
brijesh@brijesh-VirtualBox:~/Documents/ds-assign06/fibonacci$ ./fib_server  
Enter a number : 1000000000000  
Fibonacci series till 1215752192 : 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597  
2584 4181 6765 10946 17711 28657 46368 75025 121393 196418 317811 514229 832040 134626  
9 2178309 3524578 5702887 9227465 14930352 24157817 39088169 63245986 102334155 1655801  
41 267914296 433494437 701408733 1134903170  
Do you want to repeat? (y/n): n  
brijesh@brijesh-VirtualBox:~/Documents/ds-assign06/fibonacci$
```

5. Find the maximum value of an array of integers using RPC.

CODE :

```
struct maxarray{
    int size;
    int arr[100];
};

program MAXARRAY_PROG{
    version MAXARRAY_VERS{
        int max(maxarray)=1;
    }=1;
}=0x2f2f2f2f;
```

CODE - SERVER

```
/*
 * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.
 */

#include "max.h"

int *
max_1_svc(maxarray *argp, struct svc_req *rqstp)
{
    static int result;
    result = argp->arr[0];
    int n = argp->size;
    for(int i = 1 ; i < n ; i++)
    {
        if(result < argp->arr[i])
            result = argp->arr[i];
    }
    printf("Max element : %d\n", result);
    return &result;
}
```

## CODE - CLIENT

```
/*
 * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.
 */

#include "max.h"

void
maxarray_prog_1(char *host, int n)
{
    CLIENT *clnt;
    int *result_1;
    maxarray max_1_arg;

#ifdef DEBUG
    clnt = clnt_create (host, MAXARRAY_PROG, MAXARRAY_VERS, "udp");
    if (clnt == NULL) {
        clnt_pcreateerror (host);
        exit (1);
    }
#endif /* DEBUG */

    max_1_arg.size = n;
    printf("Please enter %d elements : ", n);
    for (int i = 0; i < n; i++)
        scanf("%d", &max_1_arg.arr[i]);
    result_1 = max_1(&max_1_arg, clnt);
    if (result_1 == (int *) NULL) {
        clnt_perror (clnt, "call failed");
    }
    else
    {
        printf("Largest element in the array : %d\n", *result_1);
    }
#ifdef DEBUG
```

```

        clnt_destroy (clnt);
#endif    /* DEBUG */
}

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

    if (argc < 2) {
        printf ("usage: %s server_host\n", argv[0]);
        exit (1);
    }
    host = argv[1];

    char ch='y';
    int n=0;
    while (ch=='y')
    {
        system("clear");
        printf("Please enter size of array : ");
        scanf("%d", &n);
        if(n < 1 || n>100)
        {
            while (n<1)
            {
                printf("Please enter positive integer [1-100] : ");
                scanf("%d", &n);
            }
        }
        n%=101;
        maxarray_prog_1 (host, n);
        printf("Do you want to repeat ? (y/n) : ");
        scanf("%s", &ch);
    }
    exit (0);
}

```

OUTPUT :

```
1: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/max ▾
brijesh@brijesh-VirtualBox:~/Documents/ds-assign06/max$ ./max_server
Max element : 10
█

2: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/max ▾
Please enter size of array : 10
Please enter 10 elements : 1 2 3 4 5 6 7 8 9 10
Largest element in the array : 10
Do you want to repeat ? (y/n) : █

1: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/max ▾
brijesh@brijesh-VirtualBox:~/Documents/ds-assign06/max$ ./max_server
Max element : 10
Max element : 4
█

2: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/max ▾
Please enter size of array : 12
Please enter 12 elements : 1 2 3 4 1 2 3 4 1 2 3 4
Largest element in the array : 4
Do you want to repeat ? (y/n) : █

1: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/max ▾
brijesh@brijesh-VirtualBox:~/Documents/ds-assign06/max$ ./max_server
Max element : 10
Max element : 4
Max element : 123423
█

2: brijesh@brijesh-VirtualBox: ~/Documents/ds-assign06/max ▾
Please enter size of array : 0
Please enter positive integer : -1
Please enter positive integer : 5
Please enter 5 elements : 1231 1341 123423 1345 1232
Largest element in the array : 123423
Do you want to repeat ? (y/n) : n
brijesh@brijesh-VirtualBox:~/Documents/ds-assign06/max$ █
```