Project - Assignment Bidding – Client/Server

Name – Abhishek Ranjan
UIN – 657657618

- Program takes integer input from user stating the assignment choice student wants to register
- If the assignment has seats available, Server responds with success message. Else the registration fails.
- *File* used by server for storing Assignment list and Number of seats: /assign
- *File Data Type* – AssignmentList
- *Data Type* used:

```
AssignmentList struct {
        Field1 Assignment
        Field2 Assignment
        Field3 Assignment
        Field4 Assignment
        Field5 Assignment
        Field6 Assignment
        Field7 Assignment
}
Assignment struct {
        Field0 unit64 – Id
        Field1 string – Assignment Title
        Field2 uint64 – Number of Seats Available
}
```

- *Inetrface* used:

```
My Rpc interface {
PrintAssignment () (list Assignmentlist)
BookAssignment (id uint64) (cnf uint64)
}
```

- Example of Output from Program

```
server.ethos: myRpcClient

Details For: Remote File Copy            Total Seats Left:6    (To Select, Enter 1)
Details For: Appointment Calendar        Total Seats Left:5    (To Select, Enter 2)
Details For: Password Encryption         Total Seats Left:2    (To Select, Enter 3)
Details For: Assignment Submission       Total Seats Left:5    (To Select, Enter 4)
Details For: Authorization System        Total Seats Left:7    (To Select, Enter 5)
Details For: Bell_LaPadula System        Total Seats Left:3    (To Select, Enter 6)
Details For: Approvability               Total Seats Left:5    (To Select, Enter 7)
Select choice
3
```

```
Assignment Registered

Details For: Remote File Copy            Total Seats Left:6    (To Select, Enter 1)
Details For: Appointment Calendar        Total Seats Left:5    (To Select, Enter 2)
Details For: Password Encryption         Total Seats Left:1    (To Select, Enter 3)
Details For: Assignment Submission       Total Seats Left:5    (To Select, Enter 4)
Details For: Authorization System        Total Seats Left:7    (To Select, Enter 5)
Details For: Bell_LaPadula System        Total Seats Left:3    (To Select, Enter 6)
Details For: Approvability               Total Seats Left:5    (To Select, Enter 7)
server.ethos: myRpcClient

Details For: Remote File Copy            Total Seats Left:6    (To Select, Enter 1)
Details For: Appointment Calendar        Total Seats Left:5    (To Select, Enter 2)
Details For: Password Encryption         Total Seats Left:1    (To Select, Enter 3)
Details For: Assignment Submission       Total Seats Left:5    (To Select, Enter 4)
Details For: Authorization System        Total Seats Left:7    (To Select, Enter 5)
Details For: Bell_LaPadula System        Total Seats Left:3    (To Select, Enter 6)
Details For: Approvability               Total Seats Left:5    (To Select, Enter 7)
Select choice
3

Assignment Registered

Details For: Remote File Copy            Total Seats Left:6    (To Select, Enter 1)
Details For: Appointment Calendar        Total Seats Left:5    (To Select, Enter 2)
Details For: Password Encryption         Total Seats Left:0    (To Select, Enter 3)
Details For: Assignment Submission       Total Seats Left:5    (To Select, Enter 4)
Details For: Authorization System        Total Seats Left:7    (To Select, Enter 5)
Details For: Bell_LaPadula System        Total Seats Left:3    (To Select, Enter 6)
Details For: Approvability               Total Seats Left:5    (To Select, Enter 7)
server.ethos: myRpcClient

Details For: Remote File Copy            Total Seats Left:6    (To Select, Enter 1)
Details For: Appointment Calendar        Total Seats Left:5    (To Select, Enter 2)
Details For: Password Encryption         Total Seats Left:0    (To Select, Enter 3)
Details For: Assignment Submission       Total Seats Left:5    (To Select, Enter 4)
Details For: Authorization System        Total Seats Left:7    (To Select, Enter 5)
Details For: Bell_LaPadula System        Total Seats Left:3    (To Select, Enter 6)
Details For: Approvability               Total Seats Left:5    (To Select, Enter 7)
Select choice
3

Sorry, Not able to Register the Assignment
```

- Program Listing

Server Program

```
package main

import (
        "ethos/syscall"
        "ethos/altEthos"
        "ethos/log"
        )

        var logger=log.Initialize("test/myRpcService")

        func init(){
        SetupMyRpcPrintAssignment(printAssignment)
        SetupMyRpcBookAssignment(bookAssignment)
        }

        func printAssignment()(MyRpcProcedure){

        logger.Printf("Step 3")
        logger.Printf("myRpcService:calledAssignmentlistPrint\n")
        me:=syscall.GetUser()
        path:="/user/"+me+"/assignment"
        filenm:="/assign"
        var readData Assignmentlist
        _,status:=altEthos.DirectoryOpen(path)
        if status!=syscall.StatusOk{
        logger.Printf("Error opening %v %v\n",path,status)
        }
        status=altEthos.Read(path+filenm,&readData)
        if status!=syscall.StatusOk{
        logger.Printf("Error Reading %v"+filenm+" %v\n",path,status)
        }


        return&MyRpcPrintAssignmentReply{readData}
        }

        func bookAssignment(id uint64)(MyRpcProcedure){
        logger.Printf("myRpcService:calledAssignmentbooking\n")
        logger.Printf("Step 7")
        me:=syscall.GetUser()
        path:="/user/"+me+"/assignment"
        filenm:="/assign"
        var readData Assignmentlist
        _,status:=altEthos.DirectoryOpen(path)
        if status!=syscall.StatusOk{
        logger.Printf("Error opening %v %v\n",path,status)
        }
        status=altEthos.Read(path+filenm,&readData)
        if status!=syscall.StatusOk{
        logger.Printf("Error Reading %v"+filenm+" %v\n",path,status)
        }

        datain1:=readData.Field1
        datain2:=readData.Field2
        datain3:=readData.Field3
        datain4:=readData.Field4
        datain5:=readData.Field5
        datain6:=readData.Field6
        datain7:=readData.Field7


        id1:=datain1.Field0
        id2:=datain2.Field0
        id3:=datain3.Field0
        id4:=datain4.Field0
        id5:=datain5.Field0
        id6:=datain6.Field0
        id7:=datain7.Field0
        pos1:=datain1.Field2
        pos2:=datain2.Field2
        pos3:=datain3.Field2
        pos4:=datain4.Field2
        pos5:=datain5.Field2
        pos6:=datain6.Field2
        pos7:=datain7.Field2


        var conf uint64
        logger.Printf("Input received... %v",id)

        if int(id)==int(id1)&&int(pos1)!=0{
        conf=1
        pos1=pos1-1
        }else if int(id)==int(id2)&&int(pos2)!=0{
        conf=1
        pos2=pos2-1
        }else if int(id)==int(id3)&&int(pos3)!=0{
        conf=1
        pos3=pos3-1
        }else if int(id)==int(id4)&&int(pos4)!=0{
        conf=1
        pos4=pos4-1
        }else if int(id)==int(id5)&&int(pos5)!=0{
        conf=1
        pos5=pos5-1
        }else if int(id)==int(id6)&&int(pos6)!=0{
        conf=1
        pos6=pos6-1
        }else if int(id)==int(id7)&&int(pos7)!=0{
        conf=1
        pos7=pos7-1
        }else{
        conf=99
        }
        dataout1:=Assignment{1,"Remote File Copy",pos1}
        dataout2:=Assignment{2,"Appointment Calendar",pos2}
        dataout3:=Assignment{3,"Password Encryption",pos3}
        dataout4:=Assignment{4,"Assignment Submission",pos4}
        dataout5:=Assignment{5,"Authorization System",pos5}
        dataout6:=Assignment{6,"Bell_LaPadula System",pos6}
        dataout7:=Assignment{7,"Approvability",pos7}


finaldata1:=Assignmentlist{dataout1,dataout2,dataout3,dataout4,dataout5,dataout6,datao
ut7}
        status1:=altEthos.Write(path+filenm,&finaldata1)
        if status1!=syscall.StatusOk{
        logger.Printf("Error Writing to %v"+filenm+" %v\n",path,status1)
        }

        return&MyRpcBookAssignmentReply{conf}

        }

        func main(){
        me:=syscall.GetUser()
        path:="/user/"+me+"/assignment"
        filename:="/assign"
        _,statusO:=altEthos.DirectoryOpen(path)
        if statusO!=syscall.StatusOk{
        logger.Printf("Error opening %v %v\n",path,statusO)
        }
        dataout1:=Assignment{1,"Remote File Copy",7}
        dataout2:=Assignment{2,"Appointment Calendar",5}
        dataout3:=Assignment{3,"Password Encryption",2}
        dataout4:=Assignment{4,"Assignment Submission",5}
        dataout5:=Assignment{5,"Authorization System",7}
        dataout6:=Assignment{6,"Bell_LaPadula System",3}
        dataout7:=Assignment{7,"Approvability",5}

finaldata:=Assignmentlist{dataout1,dataout2,dataout3,dataout4,dataout5,dataout6,dataout7}
        status1:=altEthos.Write(path+filename,&finaldata)
        if status1!=syscall.StatusOk{
        logger.Printf("Error Writing to %v"+filename+" %v\n",path,status1)
        }
        listeningFd,status:=altEthos.Advertise("myRpc")
        if status!=syscall.StatusOk{
        logger.Printf("Advertisingservicefailed:%s\n",status)
        altEthos.Exit(status)
        }

        for{
        _,fd,status:=altEthos.Import(listeningFd)
        if status!=syscall.StatusOk{
        logger.Printf("ErrorcallingImport:%v\n",status)
        altEthos.Exit(status)
        }
        logger.Printf("myRpcService:newconnectionaccepted\n")
        t:=MyRpc{}
        altEthos.Handle(fd,&t)
        }
        }
```

Client Program

```
package main

import (

        "ethos/altEthos"
        "ethos/syscall"
        "ethos/log"
        "ethos/kernelTypes"
        "strings"
        "strconv"
        )
        var logger=log.Initialize("test/myRpcClient")

        func init(){
        SetupMyRpcPrintAssignmentReply(printAssignmentReply)
        SetupMyRpcBookAssignmentReply(bookAssignmentReply)
        }


        func printAssignmentReply(test Assignmentlist)(MyRpcProcedure){
        logger.Printf("Step 4")
        logger.Printf("myRpcClient:ReceivedprintAssignmentReply:%v\n")

        data1:=test.Field1
        data2:=test.Field2
        data3:=test.Field3
        data4:=test.Field4
        data5:=test.Field5
        data6:=test.Field6
        data7:=test.Field7
        file1:=data1.Field1
        file2:=data2.Field1
        file3:=data3.Field1
        file4:=data4.Field1
        file5:=data5.Field1
        file6:=data6.Field1
        file7:=data7.Field1
        pos1:=data1.Field2
        pos2:=data2.Field2
        pos3:=data3.Field2
        pos4:=data4.Field2
        pos5:=data5.Field2
        pos6:=data6.Field2
        pos7:=data7.Field2

        logger.Printf("Details for "+file1+"\t \t \t \t Total Seats left:%v\n",pos1)
        logger.Printf("Details for "+file2+"\t \t \t \t Total Seats left:%v\n",pos2)
        logger.Printf("Details for "+file3+"\t \t \t \t Total Seats left:%v\n",pos3)
        logger.Printf("Details for "+file4+"\t \t \t \t Total Seats left:%v\n",pos4)
        logger.Printf("Details for "+file5+"\t \t \t \t Total Seats left:%v\n",pos5)
        logger.Printf("Details for "+file6+"\t \t \t \t Total Seats left:%v\n",pos6)
        logger.Printf("Details for "+file7+"\t \t \t \t Total Seats left:%v\n",pos7)


        str1:="\nDetails For: "+file1+"\t \tTotal Seats
Left:"+strconv.Itoa(int(pos1))+"\t (To Select, Enter 1)\n"
        str2:="Details For: "+file2+"\tTotal Seats Left:"+strconv.Itoa(int(pos2))+"\t
(To Select, Enter 2)\n"
        str3:="Details For: "+file3+"\tTotal Seats Left:"+strconv.Itoa(int(pos3))+"\t
(To Select, Enter 3)\n"
        str4:="Details For: "+file4+"\tTotal Seats Left:"+strconv.Itoa(int(pos4))+"\t
(To Select, Enter 4)\n"
        str5:="Details For: "+file5+"\tTotal Seats Left:"+strconv.Itoa(int(pos5))+"\t
(To Select, Enter 5)\n"
        str6:="Details For: "+file6+"\tTotal Seats Left:"+strconv.Itoa(int(pos6))+"\t
(To Select, Enter 6)\n"
        str7:="Details For: "+file7+"\t \tTotal Seats
Left:"+strconv.Itoa(int(pos7))+"\t (To Select, Enter 7)\n"

        var strk1 kernelTypes.String
        var strk2 kernelTypes.String
        var strk3 kernelTypes.String
        var strk4 kernelTypes.String
        var strk5 kernelTypes.String
        var strk6 kernelTypes.String
        var strk7 kernelTypes.String
        strk1=kernelTypes.String(str1)
        strk2=kernelTypes.String(str2)
        strk3=kernelTypes.String(str3)
        strk4=kernelTypes.String(str4)
        strk5=kernelTypes.String(str5)
        strk6=kernelTypes.String(str6)
        strk7=kernelTypes.String(str7)

        statusW:=altEthos.WriteStream(syscall.Stdout,&strk1)
        if statusW!=syscall.StatusOk{
        logger.Printf("Error while writing syscall.Stdout: %v",statusW)
        }
        statusW=altEthos.WriteStream(syscall.Stdout,&strk2)
        if statusW!=syscall.StatusOk{
        logger.Printf("Error while writing syscall.Stdout: %v",statusW)
        }
        statusW=altEthos.WriteStream(syscall.Stdout,&strk3)
        if statusW!=syscall.StatusOk{
        logger.Printf("Error while writing syscall.Stdout: %v",statusW)
        }
        statusW=altEthos.WriteStream(syscall.Stdout,&strk4)
        if statusW!=syscall.StatusOk{
        logger.Printf("Error while writing syscall.Stdout: %v",statusW)
        }
        statusW=altEthos.WriteStream(syscall.Stdout,&strk5)
        if statusW!=syscall.StatusOk{
        logger.Printf("Error while writing syscall.Stdout: %v",statusW)
        }
        statusW=altEthos.WriteStream(syscall.Stdout,&strk6)
        if statusW!=syscall.StatusOk{
        logger.Printf("Error while writing syscall.Stdout: %v",statusW)
        }
        statusW=altEthos.WriteStream(syscall.Stdout,&strk7)
        if statusW!=syscall.StatusOk{
        logger.Printf("Error while writing syscall.Stdout: %v",statusW)
        }

        return nil

        }

        func bookAssignmentReply(cnf uint64)(MyRpcProcedure){
        logger.Printf("Step 8")
        var str kernelTypes.String
        logger.Printf("myRpcClient:ReceivedConfirmationReply:%v\n",cnf)
        if cnf==1{
        logger.Printf("\nAssignment Registered\n");
        str="\nAssignment Registered\n"
        }else if cnf==99{
        logger.Printf("\nSorry, Not able to Register the Assignment\n");
        str="\nSorry, Not able to Register the Assignment\n"
        }
        statusW:=altEthos.WriteStream(syscall.Stdout,&str)
        if statusW!=syscall.StatusOk{
        logger.Printf("Error while writing syscall.Stdout: %v",statusW)
        }


        return nil
```

```
}

func main(){
logger.Printf("myRpcClient:beforecall\n")

//var myReader kernelTypes.String
//statusR := altEthos.ReadStream(syscall.Stdin, &myReader)
//if statusR != syscall.StatusOk {
//logger.Printf("Error while reading syscall.Stdin: %v", statusR)
//}


//statusW := altEthos.WriteStream(syscall.Stdout, &myReader)
//if statusW != syscall.StatusOk {
//logger.Printf("Error while writing syscall.Stdout: %v", statusW)
//}
fd,status:=altEthos.IpcRepeat("myRpc","",nil)
if status!=syscall.StatusOk{
logger.Printf("Ipcfailed:%v\n",status)
altEthos.Exit(status)
}

call1:=MyRpcPrintAssignment{}
status=altEthos.ClientCall(fd,&call1)
if status!=syscall.StatusOk{
logger.Printf("clientCallfailed:%v\n",status)
}

var strnew kernelTypes.String
var strnew1 kernelTypes.String
strnew="Select choice\n"
statusW:=altEthos.WriteStream(syscall.Stdout,&strnew)
if statusW!=syscall.StatusOk{
logger.Printf("Error while writing syscall.Stdout: %v",statusW)
}


statusR:=altEthos.ReadStream(syscall.Stdin,&strnew1)
if statusR!=syscall.StatusOk{
logger.Printf("Error while reading syscall.Stdin: %v",statusR)
}

//statusW = altEthos.WriteStream(syscall.Stdout, &strnew1)
//if statusW != syscall.StatusOk {
//logger.Printf("Error while writing syscall.Stdout: %v", statusW)
//}

var callvar uint64
var recvdch string

recvdch=string(strnew1)

if strings.TrimRight(recvdch,"\n")=="1"{
callvar=1
}else if strings.TrimRight(recvdch,"\n")=="2"{
callvar=2
}else if strings.TrimRight(recvdch,"\n")=="3"{
callvar=3
}else if strings.TrimRight(recvdch,"\n")=="4"{
callvar=4
}else if strings.TrimRight(recvdch,"\n")=="5"{
callvar=5
}else if strings.TrimRight(recvdch,"\n")=="6"{
```

```
callvar=6
}else if strings.TrimRight(recvdch,"\n")=="7"{
callvar=7
}else{
callvar=99
}


fd,status=altEthos.IpcRepeat("myRpc","",nil)
if status!=syscall.StatusOk{
logger.Printf("Ipcfailed:%v\n",status)
altEthos.Exit(status)
}

call2:=MyRpcBookAssignment{callvar}
status=altEthos.ClientCall(fd,&call2)
if status!=syscall.StatusOk{
logger.Printf("clientCallfailed:%v\n",status)

}

fd,status=altEthos.IpcRepeat("myRpc","",nil)
if status!=syscall.StatusOk{
logger.Printf("Ipcfailed:%v\n",status)
altEthos.Exit(status)
}

call3:=MyRpcPrintAssignment{}
status=altEthos.ClientCall(fd,&call3)
if status!=syscall.StatusOk{
logger.Printf("clientCallfailed:%v\n",status)
}


altEthos.Exit(status)
logger.Printf("myRpcClient:done\n")

}
```

MyRpc.t

```
 MyRpc interface {
PrintAssignment()(list Assignmentlist)
BookAssignment(id uint64)(cnf uint64)

}

Assignmentlist struct{
Field1 Assignment
Field2 Assignment
Field3 Assignment
Field4 Assignment
Field5 Assignment
Field6 Assignment
Field7 Assignment

}
```

```
Assignment struct{
Field0 uint64
Field1 string
Field2 uint64
}

Strm struct{
Field0 uint64
}
```