```
/*************************
  server sens
  ************************/
package main
import(
    "fmt"
    "log"
    "net/http"
    "sync"
    "os"
    "time"
//    rnd "math/rand"
//    "strconv"
)

var mu sync.Mutex
var count, count_robots, view_client int64

func main(){
  port:="localhost:8000"
  if len(os.Args)==2{
    port=os.Args[1]
  }
  println(port)
  fs:=http.FileServer(http.Dir("static"))
  http.Handle("/static/", http.StripPrefix("/static/", fs))
//  http.HandleFunc("/",find_page)
/*  http.HandleFunc("/", func(w
  http.ResponseWriter, r
  http.Request){
    http.ServeFile(w, r, "static/index.html")
}
  //http.HandleFunc("/",home_page) */
 /* errr:=http.ListenAndServe(port,nil)
  if errr !=nil{
     log.Fatal("ListenAndServei: ",errr)
  }  */
  http.HandleFunc("/",index)
  http.HandleFunc("/hand",handler)
  http.HandleFunc("/count",counter)
  http.HandleFunc("/client",client)
  http.HandleFunc("/client/list",client_list)
  http.HandleFunc("/robots.txt",robots)
  log.Fatal(http.ListenAndServe(port,nil))
}

func client_list(w http.ResponseWriter, r *http.Request){
  mu.Lock()
  view_client++
  mu.Unlock()
  http.ServeFile(w, r, "clients.lst")
}

func robots(w http.ResponseWriter, r *http.Request){
  mu.Lock()
  count_robots++
  mu.Unlock()
  http.ServeFile(w, r, "static/robots.txt")
}

func index(w http.ResponseWriter, r *http.Request){
  mu.Lock()
  count++
  mu.Unlock()
  http.ServeFile(w, r, "static/index.html")
}

func handler(w http.ResponseWriter, r *http.Request){
  mu.Lock()
```

```
  count++
  mu.Unlock()
  fmt.Fprintf(w, "%s %s %s\n", r.Method,r.URL,r.Proto)
  for k,v:=range r.Header{
      fmt.Fprintf(w, "Header[%q] = %q\n",k,v)
  }
  fmt.Fprintf(w,"Host = %q\n",r.Host)
  fmt.Fprintf(w,"RemoteAddr = %q\n",r.RemoteAddr)
  if err:=r.ParseForm();err!=nil{
      log.Print(err)
  }
  for k,v:=range r.Form{
      fmt.Fprintf(w,"Form[%q] = %q\n",k,v)
  }
}

func counter(w http.ResponseWriter, r *http.Request){
  mu.Lock()
  fmt.Fprintf(w, "Посещений: %d\nРоботов:%d\nView:%d",count,count_robots,view_client)
  mu.Unlock()
}

func client(w http.ResponseWriter,r *http.Request){
    fmt.Println("<!--start client-->")
    if err:=r.ParseForm() ;err != nil {
        log.Print(err)
    }
    var frobot,ffio,femail,fphone,faddress string
    frobot    = r.Form["product"][0]
    ffio      = r.Form["fio"][0]
    femail    = r.Form["email"][0]
    fphone    = r.Form["phone"][0]
    faddress  = r.Form["address"][0]
    if frobot != "" {
    }else{
        t:=time.Now()
        if ffio=="" && femail=="" && fphone=="" && faddress=="" {
            http.ServeFile(w, r, "static/notdocum.htm")
            /*fmt.Fprintf(w,"Пустое сообщение \n %v", t)*/
        }else{
            fmt.Printf("%v\n ФИО:   %s\n email: %s\n тел.:  %s\n адр.:  %s\n\nСчетчик:
%d\n",t,ffio,femail,fphone,faddress,count)
            fnam:="clients.lst"
            f,err:=os.OpenFile(fnam,os.O_APPEND|os.O_CREATE|os.O_WRONLY,0666)
            if err !=nil {
                 log.Fatal("not open file", err)
                 return
             }else{
                stro:=fmt.Sprintf("%v\n ФИО:   %s\n email: %s\n тел.:  %s\n адр.:
%s\nПосещений: %d\nРоботов: %d\nПросмотров:
%d\n\n",t,ffio,femail,fphone,faddress,count,count_robots,view_client)
                if _,err= f.WriteString(stro);err != nil {
                     log.Fatal("not save file", err)
                     return
                 }
                defer f.Close()
                http.ServeFile(w, r, "static/pechkin.htm")
                /*fmt.Fprintf(w,"<h2>Сообщение доставлено</h2><p>%v</p>", t)*/
            }
        }
    }

}
```