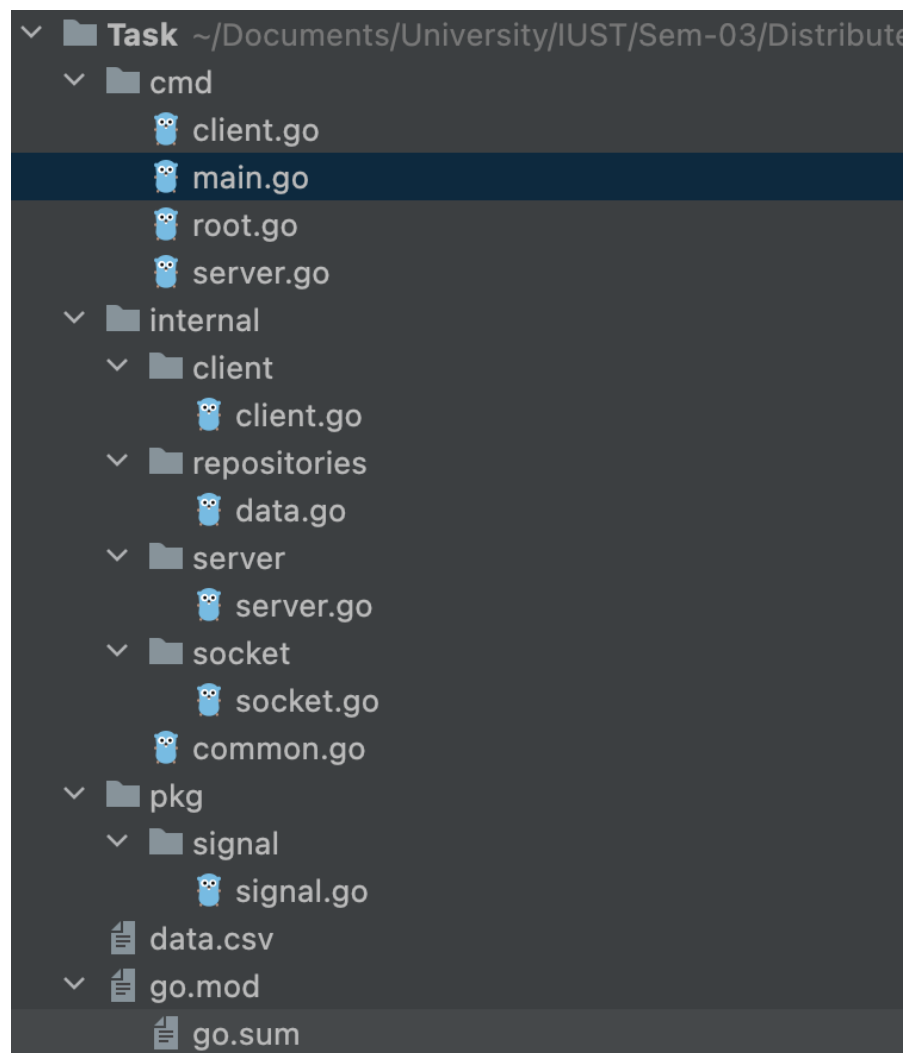


در این تمرین باید سرور و کلاینتی پیاده‌سازی می‌شد که در ابتدا کلاینت‌ها از طریق سوکت به سرور متصل می‌شدند. سپس سرور، دادگانی را به صورت تصادفی میان این کلاینت‌ها پخش می‌کرد و از این به بعد هر کدام از کلاینت‌ها برای دسترسی به هر ردیف از این دادگان باید درخواست خود را به سرور ارسال می‌کردند تا این پیام از طریق سرور به تمامی کلاینت‌ها ارسال شود و مشخص شود که مقدار داده‌ی مورد نظر چیست؟

این تمرین به زبان Go پیاده‌سازی شده است.

ساختار سورس کد پیاده‌سازی شده که همراه این گزارش پیوست شده است، به صورت زیر است:



## نحوه انجام کار و خروجی

### اجرای سرور

کامند اجرای بخش سرور به صورت زیر است:

```
go run cmd/*.go server_start
```

این کامند برای آغاز به کار سرور است. در این بخش سرور سوکتی رو بر روی یک پورت از سیستم باز می‌کند و ۱۰ ثانیه صبر می‌کند تا کلاینت‌های مختلف به آن متصل شوند. سپس دادگان را میان آن‌ها پخش می‌کند و پس از آن منتظر می‌ماند تا درخواست‌های کلاینت‌ها برای دسترسی به دادگانی که در خود کلاینت مبدا وجود ندارد را بین بقیه کلاینت‌ها پخش کند و نتیجه را به کلاینت مبدا باز گرداند.

```
2  ~/Doc/University/IUST/Sem-03/Distributed Systems/HW1/Task  go run cmd/*.go server_start
Server Started.
>> Loading data...
      10000 rows of data added.
>> Listening on 127.0.0.1:9080
>> Waiting 10 seconds for clients to connect...
      Client [0]: 127.0.0.1:55938 connected.
      Client [1]: 127.0.0.1:55939 connected.
      Client [2]: 127.0.0.1:55940 connected.
      Client [3]: 127.0.0.1:55941 connected.
>> Start distributing data...
      Sending 4500 rows of data to client 0
      Sending 2000 rows of data to client 1
      Sending 200 rows of data to client 2
      Sending 3300 rows of data to client 3
      Data distributed among clients.
Data of ID '109992' requested from 1 found in 0
Data of ID '100009' requested from 1 found in 0
Data of ID '109992' requested from 2 found in 0
Data of ID '100009' requested from 2 found in 0
Data of ID '109992' requested from 3 found in 0
Data of ID '100009' requested from 3 found in 0
```

همان طور که در تصویر هم مشاهده می‌شود، ابتدا سرور آغاز به کار کرده. پس از آن ۱۰ هزار داده‌ی مورد نظر توسط آن خوانده شده. سپس ۱۰ ثانیه منتظر اضافه شدن سایر کلاینت‌ها بوده که برای هر کدام از آن‌ها سوکتی برای تبادل پیام ساخته شود. در نهایت داده‌ها به صورت تصادفی و همچنین تعداد تصادفی بین کلاینت‌های مختلف پخش شده است.

در خطوط پایانی تصویر هم مشاهده می‌شود که درخواست‌هایی از کلاینت‌های مختلف وجود داشته که توسط یکی دیگر از کلاینت‌ها که صاحب آن داده بوده به سرور داده شده تا به کلاینت مبدا تحویل داده شود.

## اجرای کلاینت

کامند مورد نیاز برای شروع هر کدام از کلاینت‌ها نیز به صورت زیر است:

```
go run cmd/*.go client_start
```

این برنامه توانایی اتصال هر تعداد کلاینت را دارد یعنی هر تعداد کلاینت که در ۱۰ ثانیه‌ی ابتدایی به سرور متصل شود برای تقسیم داده‌ها در نظر گرفته می‌شود. در این مثال با ۴ کلاینت تست، اجرای برنامه صورت گرفته است. در تصاویر زیر اجرای هر کدام از کلاینت‌ها را به ترتیب مشاهده می‌کنیم.

```
z ~ /Doc/U/IUST/Sem-03/Distributed Systems/HW1/Task go run cmd/*.go client_start
>> Client Started.
>> Connected to server.
>> Waiting for data...
    Start receiving data
    End of receiving data
    4500 rows of data added.
109992,100009
    Searching for ID 109992...
* Data of ID '109992' is : {'109992' 'Carlos' 'D'"Amore-Lang' 'Carlos.DAmore-Lang44@hotmail.com' 'Rociotown'}
    Searching for ID 100009...
* Data of ID '100009' is : {'100009' 'Wendell' 'Krajcik-Breitenberg' 'Wendell41@yahoo.com' 'Arjunshire'}
```

به کلاینت اول ۴۵۰۰ داده تحویل شده. پس از آن ۲ آیدی از آن خواسته شده که کلاینت متوجه نمی‌شود از کجا این داده به آن تحویل داده شده ولی در تصویر قبلی که مربوط به سرور بود مشخص است که این داده درون همین کلاینت وجود داشته و از سایر کلاینت‌ها گرفته نشده. در حقیقت Transparency رعایت شده است.

در ۳ کلاینت دیگر اما درخواست‌ها به سرور رفته و سرور داده‌ها را از کلاینت ۱ گرفته و به سایر کلاینت‌ها داده است.

```
z ~ /Doc/University/IUST/Sem-03/Distributed Systems/HW1/Task go run cmd/*.go client_start
>> Client Started.
>> Connected to server.
>> Waiting for data...
    Start receiving data
    End of receiving data
    2000 rows of data added.
109992,100009
    Searching for ID 109992...
    Searching for ID 100009...
* Data of ID '109992' is : {'109992' 'Carlos' 'D'"Amore-Lang' 'Carlos.DAmore-Lang44@hotmail.com' 'Rociotown'}
* Data of ID '100009' is : {'100009' 'Wendell' 'Krajcik-Breitenberg' 'Wendell41@yahoo.com' 'Arjunshire'}
```

```
2  ~/.Doc/U/IUST/Sem-03/Distributed Systems/HW1/Task  go run cmd/*.go client_start
>> Client Started.
>> Connected to server.
>> Waiting for data...
    Start receiving data
    End of receiving data
    200 rows of data added.
109992,100009
    Searching for ID 109992...
    Searching for ID 100009...
* Data of ID '109992' is : {'109992' 'Carlos' 'D'"Amore-Lang' 'Carlos.DAmore-Lang44@hotmail.com' 'Rociotown'}
* Data of ID '100009' is : {'100009' 'Wendell' 'Krajcik-Breitenberg' 'Wendell41@yahoo.com' 'Arjunshire'}
```

```
2  ~/.Doc/U/IUST/Sem-03/Distributed Systems/HW1/Task  go run cmd/*.go client_start
>> Client Started.
>> Connected to server.
>> Waiting for data...
    Start receiving data
    End of receiving data
    3300 rows of data added.
109992,100009
    Searching for ID 109992...
    Searching for ID 100009...
* Data of ID '109992' is : {'109992' 'Carlos' 'D'"Amore-Lang' 'Carlos.DAmore-Lang44@hotmail.com' 'Rociotown'}
* Data of ID '100009' is : {'100009' 'Wendell' 'Krajcik-Breitenberg' 'Wendell41@yahoo.com' 'Arjunshire'}
```