



## مقدمه

این پروژه به جمع‌بندی آموخته‌های شما در درس می‌پردازد. انتظار می‌رود مهارت‌هایی را که در تمرین‌های پیشین و سایر بخش‌های درس آموخته‌اید در پیاده‌سازی این پروژه به کار گیرید.

شما باید شبکه‌ای مجازی پیاده‌سازی کنید که در آن افراد می‌توانند به با دو نقش مختلف در آن ثبت‌نام کنند و از امکاناتی که در اختیار آن‌ها قرار داده می‌شود استفاده کنند. نقش اول ناشر است که می‌تواند در شبکه‌ی مجازی فیلم‌های خود را قرار دهد تا بقیه‌ی افراد بتوانند آن‌ها را خریداری کرده و مشاهده کنند. نقش دوم نیز مشتری‌ها هستند که می‌توانند به صورت جداگانه در شبکه ثبت‌نام کنند و به مشاهده و امتیازدهی فیلم‌ها بپردازند.

در فاز اول شما برنامه‌ی این شبکه‌ی مجازی را پیاده‌سازی می‌کنید که با استفاده از رابط کاربری خط فرمان<sup>۱</sup> می‌توان از آن استفاده کرد. در فازهای بعدی این برنامه گسترش می‌یابد.

## ۱ شرح تمرین

در این قسمت تعدادی دستور در ادامه تعریف می‌شود که شما باید آن‌ها را پیاده‌سازی کنید. تمامی دستورها پس از اجرا شدن دارای خروجی مشخصی هستند که منحصراً ذکر می‌شود. دقت کنید که خروجی‌های شما به صورت خودکار تست می‌شوند. لذا خروجی شما باید دقیقاً همانند خروجی خواسته شده باشد و در غیر این صورت نمره‌ی بخش آزمون<sup>۲</sup> را از دست خواهید داد.

## ۲ خطاها

در این پروژه شما تنها ملزم به بررسی سه نوع خطای زیر و خروجی دادن آن هستید که در ادامه آن‌ها را معرفی می‌کنیم.

### ۱.۲ خطای عدم وجود

اگر دستوری که کاربر وارد می‌کند (قسمت قبل از ؟ در دستورها) در لیست دستورهای تعریف شده نباشد یا کاربر با وارد کردن شناسه‌ی ناموجود سیستم قصد دسترسی به موجودیتی را داشته باشد این خطا در خروجی چاپ می‌شود.

خروجی

Not Found

### ۲.۲ خطای درخواست اشتباه

اگر اولین قسمت خط ورودی کاربر یکی از چهار مقدار POST، PUT، GET یا DELETE نباشد یا این دستور اطلاعات کافی برای اجرا را در خود نداشته باشد سیستم خروجی زیر را چاپ می‌کند.

خروجی

Bad Request

<sup>1</sup>Command Line Interface

<sup>2</sup>testcase

## ۳.۲ خطای عدم دسترسی

شرایط رخداد این دستور در ادامه‌ی متن آمده است. در صورت رخداد این خطا پیام زیر چاپ می‌شود.

خروجی  
Permission Denied

## ۳ دستورها

### ۱.۳ ثبت‌نام

با این دستور افراد می‌توانند در شبکه‌ی مجازی ثبت‌نام کنند. نام کاربری هرکاربر یک رشته‌ی یکتا در سیستم است. توجه کنید که هر کاربر دارای یک شناسه‌ی عددی یکتا است که این شناسه از عدد یک شروع می‌شود. این شناسه به طور خودکار در هنگام ثبت‌نام کاربر به آن تعلق می‌گیرد. اگر نام کاربری در سیستم موجود باشد درخواست معتبر نیست و خروجی Bad Request چاپ می‌شود.

وجود ویژگی ناشربودن اختیاری است و در صورت ذکر نشدن نادرست فرض می‌شود.

دستور ورودی  
POST signup ? email <email> username <username> password <password> age <age> [publisher <true|false>]

خروجی  
OK

### ۲.۳ ورود

اگر فردی قبلاً در شبکه ثبت‌نام کرده باشد، پیش از استفاده از امکانات سامانه باید در آن وارد شود. توجه کنید که کاربران بلافاصله بعد از ثبت‌نام نیازی به دستور ورود ندارند و پس از ثبت‌نام به صورت ضمنی با همان دستور ثبت‌نام وارد سیستم نیز شده است. در صورت درست نبودن اطلاعاتی که کاربر وارد کرده است پیام Bad Request در خروجی چاپ شود.

دستور ورودی  
POST login ? username <username> password <password>

خروجی  
OK

### ۳.۳ ناشران

دو دستور قبل برای آن بود که کاربرها بتوانند به منظور استفاده از امکانات به سامانه وارد شوند. هر نوع از کاربران قابلیت‌های خاص خود را دارند که از این پس قابلیت‌های آن‌ها را در قالب دستورهایی بیان می‌کنیم.

اگر کاربری به سامانه وارد نشده باشد یا دستوری را صدا بزند که مربوط به نوع کاربری او نیست باید پیام خطا Permission Denied داده شود.

### ۱.۳.۳ ثبت فیلم

هر ناشر می‌تواند با دستور زیر اطلاعات فیلم خود را وارد کرده و آن را به شبکه اضافه کند. کاربرد قیمت فیلم‌ها آن است که هر کاربری که بخواهد این فیلم را مشاهده کند باید معادل قیمتش را پرداخت کند.

توجه کنید که هر فیلم دارای یک شناسه‌ی عددی یکتا است که این شناسه از عدد یک شروع می‌شود و در صورت حذف شدن یک فیلم جایگزین نمی‌شود. این شناسه به طور خودکار در هنگام اضافه شدن فیلم به آن تعلق می‌گیرد.

دستور ورودی
POST films ? name <name> year <year> length <length> price <price> summary <summary> director <director>
خروجی
OK

### ۲.۳.۳ ویرایش اطلاعات فیلم

هر ناشر می‌تواند اطلاعات فیلم خود را ویرایش کند. در این دستور کاربر باید شناسه‌ی فیلم را وارد کند و سپس هر بخشی از اطلاعات فیلم را که می‌خواهد تغییر بدهد جلوی آن بیاورد. در صورتی که فیلم مورد نظر به کاربر درخواست‌کننده تعلق نداشت پیام Permission Denied در خروجی داده می‌شود.

دستور ورودی
PUT films ? film_id <film_id> [name <name>] [year <year>] [length <length>] [summary <summary>] [director <director>]
خروجی
OK

### ۳.۳.۳ حذف فیلم

هر ناشر می‌تواند فیلمی را که خودش اضافه کرده است با دستور زیر از شبکه حذف کند. در صورت عدم وجود فیلم مورد نظر در سیستم پیام Not Found و در صورتی که فیلم مورد نظر به کاربر درخواست‌کننده تعلق نداشت پیام Permission Denied در خروجی داده می‌شود.

دستور ورودی
DELETE films ? film_id <film_id>
خروجی
OK

### ۴.۳.۳ مشاهده لیست دنبال‌کنندگان

هر ناشر می‌تواند لیست تمامی کسانی را که او را دنبال<sup>۳</sup> می‌کنند با دستور زیر مشاهده کند. این نتایج باید بر حسب شناسه‌ی کاربرها مرتب شده باشد.

<sup>3</sup>follow

دستور ورودی

---

GET followers

---

خروجی

---

List of Followers

#.	User Id	User Username	User Email
1.	<user_id>	<user_username>	<user_email>
2.	<user_id>	<user_username>	<user_email>
.			
.			
.			
#n.	<user_id>	<user_username>	<user_email>

---

### ۵.۳.۳ دریافت پول از شبکه

هر ناشر می‌تواند با دستور زیر پولی را که از فروش فیلم‌هایش بدست آورده است از شبکه دریافت کند. هر فیلم در شبکه ممکن است یکی از سه حالت زیر را داشته باشد:

۱. ضعیف: اگر امتیاز یک فیلم از ۵ کمتر باشد، این فیلم در حالت ضعیف قرار دارد.

۲. متوسط: اگر امتیاز یک فیلم بیشتر مساوی ۵ باشد و کمتر از ۸ باشد، این فیلم در حالت متوسط قرار دارد.

۳. قوی: اگر امتیاز یک فیلم بیشتر مساوی ۸ باشد، این فیلم در حالت قوی قرار دارد.

هنگامی که کاربری پولی برای یک فیلم پرداخت می‌کند، بر اساس اینکه فیلم در کدام یک از حالت‌های بالا باشد، مقداری از آن به ناشر تعلق می‌گیرد و باقی آن در حساب شبکه باقی می‌ماند. هر ناشر برای فیلم خود در حالت ضعیف باید ۸۰ درصد، در حالت متوسط ۹۰ درصد و در حالت قوی ۹۵ درصد از مبلغ را از شبکه دریافت کند. توجه کنید که میزان پرداخت به ناشر دقیقاً به امتیاز فیلم در لحظه‌ی خرید بستگی دارد. یعنی اگر در هنگام دریافت پول امتیاز فیلم تغییر کرده بود، میزان پول دریافتی تغییری نمی‌کند و تنها امتیاز فیلم در هنگام خرید مبنای محاسبه‌ی دریافتی‌ها است.

دستور ورودی

---

POST money

---

خروجی

---

OK

---

### ۶.۳.۳ مشاهده‌ی لیست فیلم‌های منتشر کرده

هر ناشر می‌تواند با دستور زیر فیلم‌هایی را که خودش در سایت قرار داده است مشاهده کند.

دستور ورودی  
GET published ? [name <name>] [min\_rate <min\_rate>] [min\_year <min\_year>] [price <price>] [max\_year <max\_year>] [director <director>]

خروجی  
#. Film Id | Film Name | Film Length | Film price | Rate | Production Year | Film Director  
1. <film\_id> | <film\_name> | <film\_length> | <Film price> | <Rate> | <Production Year> | <film\_director>  
2. <film\_id> | <film\_name> | <film\_length> | <Film price> | <Rate> | <Production Year> | <film\_director>  
.  
.  
.  
#n. <film\_id> | <film\_name> | <film\_length> | <Film price> | <Rate> | <Production Year> | <film\_director>

### ۷.۳.۳ پاسخ دادن به نظرات

هر ناشر می‌تواند به نظرانی که به فیلم‌هایش داده شده پاسخ دهد. در صورت موفقیت‌آمیز بودن این دستور یک پیام به کاربر نویسنده‌ی نظر ارسال می‌شود.

دستور ورودی  
POST replies ? film\_id <film\_id> comment\_id <comment\_id> content <content>

خروجی  
OK

### ۸.۳.۳ حذف کردن نظرات

هر ناشر می‌تواند نظرانی را که به فیلم‌هایش داده شده حذف کند.

دستور ورودی  
DELETE comments ? film\_id <film\_id> comment\_id <comment\_id>

خروجی  
OK

### ۴.۳ مشتریان

تا این جا با قابلیت‌هایی که تنها ناشر می‌تواند انجام دهد، آشنا شدیم. از این پس با قابلیت‌های مشتری‌ها آشنا می‌شویم. توجه کنید که هر ناشر برای فیلم‌های دیگر می‌تواند در نقش یک مشتری باشد. یعنی تمام قابلیت‌های مشتری را نیز علاوه بر قابلیت‌های ناشر داراست.

### ۱.۴.۳ دنبال کردن یک ناشر

هر مشتری می‌تواند با دستور زیر ناشر مورد علاقه‌ی خود را دنبال کند. دنبال کردن ناشر این ویژگی را دارد که وقتی این ناشر فیلم جدیدی قرار می‌دهد به تمامی مشتری‌هایی که آن را دنبال کرده‌اند، اطلاع‌رسانی می‌شود. همچنین هنگامی که یک ناشر توسط کاربری دنبال می‌شود باید برای آن ناشر پیغامی<sup>۴</sup> ارسال شود.

```
----- دستور ورودی -----
POST followers ? user_id <user_id>
-----
----- خروجی -----
OK
-----
```

### ۲.۴.۳ اضافه کردن اعتبار

هر مشتری می‌تواند حساب خود را شارژ کند و پول خود را افزایش دهد.

```
----- دستور ورودی -----
POST money ? amount <amount>
-----
----- خروجی -----
OK
-----
```

### ۳.۴.۳ جستجو

هر مشتری می‌تواند با دستور زیر بین تمامی فیلم‌ها جستجو انجام دهد. در این دستور ممکن است هر ویژگی از ویژگی‌های فیلم قرار گیرد که در صورت وجود باید فیلم‌ها برآن اساس فیلتر شود. در صورت وجود چند ویژگی باید نتایج نهایی شامل تمامی فیلترهای خواسته شده باشد. اگر هیچ ویژگی‌ای وارد نشود، خروجی باید شامل تمامی فیلم‌ها باشد.

```
----- دستور ورودی -----
GET films ? [name <name>] [min_rate <min_rate>] [min_year <min_year>] [price <price>] [max_year <max_year>] [director <director>]
-----
----- خروجی -----
#. Film Id | Film Name | Film Length | Film price | Rate | Production Year | Film Director
1. <film_id> | <film_name> | <film_length> | <Film price> | <Rate> | <Production Year> | <film_director>
2. <film_id> | <film_name> | <film_length> | <Film price> | <Rate> | <Production Year> | <film_director>
.
.
.
#n. <film_id> | <film_name> | <film_length> | <Film price> | <Rate> | <Production Year> | <film_director>
-----
```

<sup>4</sup>notification

### ۴.۴.۳ مشاهده جزئیات فیلم

هر مشتری می‌تواند با دستور زیر جزئیات یک فیلم را مشاهده کند. همچنین در این قسمت پس از جزئیات باید نظرهایی را که برای آن فیلم گذاشته شده و پاسخ‌هایی را که آن نظرات دریافت کرده‌اند نمایش داده شود. در انتهای این قسمت ۴ فیلم که بالاترین امتیاز را در میان فیلم‌های شبکه دارند پس از نمایش جزئیات فیلم مذکور به نمایش در می‌آید. اگر امتیاز آن‌ها برابر بود ترتیب اضافه‌شدن معیار انتخاب خواهد بود و از بین گزینه‌های برابر قدیمی‌ترین‌ها انتخاب می‌شوند. توجه کنید که این فیلم‌ها نباید جزء فیلم‌هایی باشد که این کاربر قبلاً خریداری کرده است.

دستور ورودی	
GET films ? film_id <film_id>	
خروجی	
Details of Film <film_name>	
Id = <film_id>	
Director = <film_director>	
Length = <film_length>	
Year = <film_year>	
Summary = <film_summary>	
Rate = <rate>	
Price = <price>	
Comments	
<comment-Id>. <comment_message>	
<comment-id>.1. <reply_message>	
Recommendation Film	
#. Film Id   Film Name   Film Length   Film Director	
1. <film_id>   <film_name>   <film_length>   <film_director>	
2. <film_id>   <film_name>   <film_length>   <film_director>	
3. <film_id>   <film_name>   <film_length>   <film_director>	
4. <film_id>   <film_name>   <film_length>   <film_director>	

### ۵.۴.۳ خرید فیلم

هر مشتری می‌تواند با دستور زیر پس از پرداخت هزینه‌ی یک فیلم آن را خریداری کند. این هزینه مستقیماً به دارایی شبکه‌ی مجازی اضافه خواهد شد. پس از خرید فیلم یک پیام برای ناشر فیلم ارسال می‌شود.

دستور ورودی	
POST buy ? film_id <film_id>	
خروجی	
OK	

### ۶.۴.۳ امتیازدهی به فیلم‌ها

هر مشتری می‌تواند پس از آن که فیلمی را خرید با دستور زیر بین ۱ تا ۱۰ به آن امتیاز دهد. امتیاز هر فیلم از میانگین امتیازاتی که مشتری‌ها به آن‌ها می‌دهند، محاسبه می‌شود. پس از هر امتیاز دادن پیغامی به ناشر آن فیلم ارسال می‌شود.

دستور ورودی

POST rate ? film\_id <film\_id> score <score>

خروجی

OK

### ۷.۴.۳ نظردهی به فیلم‌ها

هر مشتری می‌تواند پس از آن که فیلمی را خرید نظر خود را زیر آن فیلم ثبت کند. پس از هر نظر دادن پیغامی حاوی این پیام به ناشر ارسال می‌شود.

توجه کنید که هر نظر دارای یک شناسه‌ی عددی یکتا است که این شناسه برای هر فیلم به صورت جداگانه از عدد یک شروع می‌شود. این شناسه به طور خودکار در هنگام اضافه شدن نظر به آن تعلق می‌گیرد.

دستور ورودی

POST comments ? film\_id <film\_id> content <content>

خروجی

OK

### ۸.۴.۳ مشاهده‌ی لیست فیلم‌های خریداری شده

هر مشتری می‌تواند لیست فیلم‌هایی را که خریداری کرده است مشاهده کند. این لیست باید بر حسب شناسه‌ی فیلم‌ها مرتب شده باشد.

دستور ورودی

GET purchased ? [name <name>] [price <price>] [min\_year <min\_year>] [max\_year <max\_year>] [director <director>]

خروجی

#. Film Id | Film Name | Film Length | Film price | Rate | Production Year | Film Director

1. <film\_id> | <film\_name> | <film\_length> | <Film price> | <Rate> | <Production Year> | <film\_director>

2. <film\_id> | <film\_name> | <film\_length> | <Film price> | <Rate> | <Production Year> | <film\_director>

.

.

.

#n. <film\_id> | <film\_name> | <film\_length> | <Film price> | <Rate> | <Production Year> | <film\_director>



### ۹.۴.۳ مشاهده لیست پیغام‌های خوانده نشده

هر مشتری می‌تواند لیست پیغام‌هایی را که در زمان عدم حضور او در حساب کاربری ارسال شده است با این دستور مشاهده کند. پس از آن که هر فرد پیغام‌های خوانده نشده خود را مشاهده کرد این پیغام‌ها از لیست پیغام‌های خوانده نشده حذف می‌شود. این لیست باید به ترتیب رسیدن پیغام‌ها مرتب شده باشد؛ یعنی جدیدترین پیغام باید در اول لیست نشان داده شود.

دستور ورودی	
GET notifications	
خروجی	
#. Notification Message	
1. <notification_message>	
2. <notification_message>	
.	
.	
.	
#n. <notification_message>	

### ۱۰.۴.۳ مشاهده لیست همه‌ی پیغام‌ها

در این قسمت هر کاربر می‌تواند تمامی پیغام‌هایی را که تا به حال برای او آمده است مشاهده کند. این پیغام‌ها فقط شامل پیغام‌های خوانده شده است. همچنین در این قسمت حدی برای مشاهده‌ی پیغام‌ها وجود دارد که باید تا آن تعداد پیغام به نمایش درآید. توجه کنید که مانند بالا این لیست باید به ترتیب رسیدن پیغام‌ها مرتب شده باشد؛ یعنی جدیدترین پیغام باید در اول لیست نشان داده شود.

دستور ورودی	
GET notifications read ? limit <limit>	
خروجی	
#. Notification Message	
1. <notification_message>	
2. <notification_message>	
.	
.	
.	
#n. <notification_message>	

فرمت خروجی پیغام‌ها دارای انواع زیر است.

1. Publisher <publisher\_name> with id <publisher\_id> reply to your comment.
2. Publisher <publisher\_name> with id <publisher\_id> register new film.
3. User <user\_name> with id <user\_id> follow you.
4. User <user\_name> with id <user\_id> buy your film <film\_name> with id <film\_id>.
5. User <user\_name> with id <user\_id> rate your film <film\_name> with id <film\_id>.
6. User <user\_name> with id <user\_id> comment on your film <film\_name> with id <film\_id>.

## ۴ بخش امتیازی

### ۱.۴ git

در این پروژه، برای کسب امتیاز بیشتر، در سایت GitHub<sup>۵</sup> عضو شده و یک مخزن<sup>۶</sup> خصوصی<sup>۷</sup> جدید ایجاد کنید. سپس پروژه را با استفاده از git توسعه دهید. پس از اعمال هر تغییر، فایل جدید را با پیامی مناسب که نشانگر تغییرات اعمال شده در این نسخه نسبت به نسخه قبلی است commit و سپس push کنید. همه‌ی commitها را روی شاخه‌ی<sup>۸</sup> اصلی<sup>۹</sup> push کنید. توجه کنید که هر commit باید تنها شامل یک تغییر باشد.

دقت کنید که در هنگام تحویل commitهای شما بررسی می‌شود و تنها در صورت استفاده‌ی بهینه و درست از git نمره‌ی امتیازی آن به شما تعلق می‌گیرد. یعنی تنها ساختن یک مخزن و پوش کردن پروژه در آن نمره‌ای نخواهد داشت.

### ۲.۴ هش کردن گذرواژه

همان‌طور که می‌دانید گذرواژه ویژگی‌ای است که افراد تمایل دارند از دسترسی سایرین حفظ شود. به این دلیل است که هنگام ذخیره‌سازی گذرواژه آن را به صورت خام در حافظه نگهداری نمی‌کنند. هش<sup>۱۰</sup> کردن روشی که توسط آن می‌توان یک رشته را طی فرآیندی غیرقابل بازگشت به رشته‌ی دیگری تبدیل کرد؛ یعنی هش هر رشته یک رشته‌ی یکتا است و با هر رشته‌ی دیگر تفاوت دارد و همچنین از طریق هش یک رشته نمی‌توان به آن رشته دسترسی پیدا کرد.

در این قسمت باید گذرواژه‌های افراد را به صورت هش شده در حافظه نگهداری کنیم تا از دسترسی به گذرواژه خام پرهیز شود. هش کردن الگوریتم‌های متفاوتی دارد که می‌توانید در اینترنت درباره‌ی آن‌ها جستجو کنید.

## ۵ نکات پایانی

○ در تخصیص شناسه‌ها به موجودیت‌های سیستم دقت کنید. تمام شناسه‌ها به صورت خودکار و از عدد یک شروع می‌شوند و در صورت حذف شدن یک شناسه مقدار آن جایگزین نمی‌شود؛ حتی اگر مقدار حذف شده آخرین مقدار تخصیص داده شده باشد.

<sup>۵</sup>[github.com](https://github.com)

<sup>۶</sup>repository

<sup>۷</sup>private

<sup>۸</sup>branch

<sup>۹</sup>master

<sup>۱۰</sup>hash

## ۶ نحوه‌ی تحویل

پرونده‌های مربوط به برنامه‌ی خود را در پوشه‌ای با نام A7-1-SID.zip در صفحه‌ی CECM درس بارگذاری کنید که SID شماره‌ی دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۷۹۹۹ باشد، نام پوشه‌ی شما باید A7-1-810197999.zip باشد.

- برنامه‌ی شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++11 ترجمه و اجرا شود.
- برنامه‌ی شما باید حتماً طراحی شیء‌گرا داشته باشد. همچنین باید به صورت Multifile باشد و استفاده از Makefile در این تمرین اجباری است.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.