



تمرین برنامه‌نویسی صفرم^۱ شبکه‌های کامپیوتری

مدرس: مهدی خرازی

زمستان ۱۳۹۰

در این تمرین شما باید یک بازی را به صورت Client/Server با استفاده از مفاهیم Socket در محیط Unix به زبان C یا C++ پیاده‌سازی کنید.

هدف

آشنایی با

- محیط سیستم‌عامل‌های UNIX
- محیط برنامه‌نویسی Eclipse
- UNIX Socket API
- POSIX Thread API
- NCURSES Library
- نوشتن Makefile
- ساختن پوشه، نام‌گذاری و فشرده‌سازی و ارسال صحیح تمرین

مقدمه

برای انجام دادن بازی روی شبکه، تمهیداتی باید اندیشیده شود که به طور عادی برای بازی کردن تنها روی یک دستگاه با آن‌ها مواجه نمی‌شویم. مهم‌ترین مسئله‌ای که ایجاد می‌شود هماهنگ کردن بازیکن‌های مختلف با همدیگر است. بازیکن‌ها باید به طور منصفانه‌ای از اطلاعات بازی مطلع شوند و کارگزار باید بتواند به طور همزمان به حرکت‌های بازیکن‌های مختلف رسیدگی کند و منابع مشترک بین بازیکن‌ها را به درستی مدیریت کند. در ادامه با طراحی یک بازی روی شبکه، تمهیدات گفته شده را برای پیاده‌سازی، بررسی می‌کنیم.

^۱ با تشکر از بهنام مومنی، مهدی احمدی‌نژاد، کامیار اللهوردی، سجاد فولادی، علی محمد ربانی، مهرداد مرادی و بهنام حاتمی

برنامه‌ی کارگزار

این برنامه نقش مدیریت بازی را اجرا می‌کند. بازیکن‌ها برای بازی کردن باید به کارگزار وصل شوند. برنامه‌ی کارگزار برای اجرا شدن به یک نقشه برای بازی و پورته‌ی که باید روی آن منتظر متصل شدن بازیکن‌ها باشد، نیاز دارد:

```
$ ./server <port> <map-file>
```

برای مثال برای اجرای کارگزار روی پورت ۸۰۸۰ با نقشه‌ای که در فایل map.txt وجود دارد، چنین دستوری را باید وارد کنیم:

```
$ ./server 8080 map.txt
```

فایل حاوی نقشه به صورت زیر ساخته می‌شود:

```
<rows>
<cols>
<celltype1-1> <celltype1-2>... <celltype1-<cols>>
<celltype2-1> <celltype2-2>... <celltype2-<cols>>
...
<celltype<rows>-1> <celltype<rows>-2>... <celltype<rows>-<cols>>
```

مقادیر <celltype-i-j> یکی از مقادیر زیر می‌تواند باشد:

Type	Celltype Value
Empty	E
Gold	G
Coin	C
Red Player	R
Blue Player	B

برای نمونه فایل معادل برای نقشه‌ی زیر در ادامه آورده شده است:



فایل معادل این نقشه:

```
7
6
EEEEEE
ECEERE
EEEEEE
EEECCE
EGEEEC
EEEEEE
EBEEEE
```

شروع بازی

کارگزار پس از خواندن نقشه‌ی مرتبط با بازی، روی پورت تعیین شده منتظر وصل شدن بازیکن‌ها می‌ماند. در این بازی تنها دو بازیکن آبی و قرمز داریم که کارگزار باید منتظر وصل شدن آن‌ها بماند. هر کدام از بازیکن‌ها پس از اتصال باید با ارسال یک رشته، آبی یا قرمز بودن خود را مشخص کنند. بازیکن قرمز باید رشته‌ی Red و بازیکن آبی باید رشته‌ی Blue را ارسال کند.

<Red-or-Blue>

برای نمونه بازیکن قرمز پس از وصل شدن به کارگزار، باید پیغام زیر را برای کارگزار ارسال کند:

Blue

کارگزار باید منتظر پیغام اولیه دو بازیکن بماند. پس از دریافت پیغام اولیه از دو بازیکن، کارگزار بازی را باید طبق قسمت بعد آغاز کند.

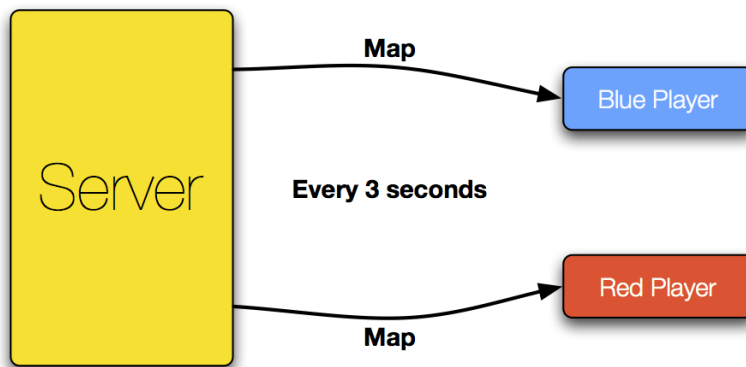
در جریان بازی

پس از اتصال دو بازیکن، کارگزار با ارسال نقشه برای هر دو بازیکن منتظر حرکت آنها می‌ماند. تمامی اطلاعات ارسالی بین بازیکن‌ها و کارگزار باید با کاراکتر Null خاتمه یافته باشد.

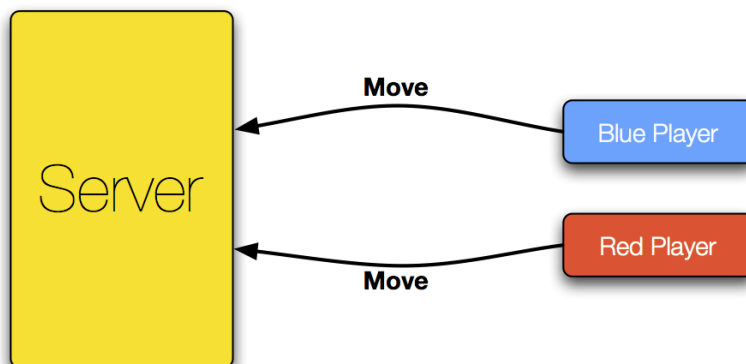
```
0
EEEEEE
ECEERE
EEEEEE
EEECCE
EGEEEC
EEEEEE
EBEEEE
```

کارگزار علاوه بر اطلاعات نقشه، امتیاز هر بازیکن را نیز در ابتدای اطلاعات ارسالی قرار می‌دهد.

ارسال این اطلاعات برای بازیکن‌ها باید هر ۳ ثانیه یکبار توسط کارگزار انجام پذیرد.^۲

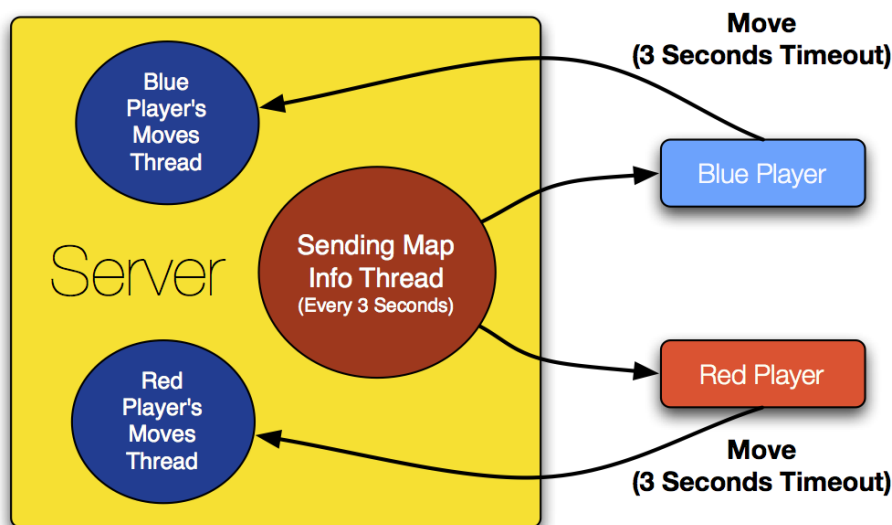


در مقابل کارگزار باید حرکت‌های انجام شده توسط بازیکن‌ها را دریافت کند و با توجه به حرکت انجام شده، نقشه‌ی بازی به همراه امتیاز بازیکن‌ها را به‌روز کند.



^۲ اعداد ثابت عنوان شده، مانند ارسال ۳ ثانیه یکبار اطلاعات نقشه توسط کارگزار، را باید به صورت [ماکرو](#) تعریف کنید.

توجه داشته باشید که کارگزار باید به طور همزمان قادر به دریافت و پردازش حرکتهای بازیکنها باشد. به عبارتی دیگر برای دریافت حرکتهای هر بازیکن باید یک ریسهی مستقل ایجاد کند. ساختار ریسههای کارگزار به شکل زیر است:



همچنین کارگزار باید دو نوع حرکت را از سوی بازیکنها نپذیرد:

۱. حرکتی که منجر به خروج بازیکن از صفحه شود.
۲. حرکتی که منجر به قرارگیری یک بازیکن روی خانهی بازیکن دیگر شود.

بازیکن باید با پردازش اطلاعات دریافتی و نمایش آن برای برنامهی بازیکن، مشابه با آنچه در کارگزار برای نمایش اطلاعات بازی وجود دارد، اطلاعات بازی را در هر لحظه نمایش دهد. نقشه به همراه امتیاز بازیکن باید نمایش داده شود.

کارگزار پس از انتظار حداکثر به مدت ۳ ثانیه برای هر بازیکن، حرکت ارسال شده توسط آن بازیکن را باید اجرا کند. در صورتی که حرکتی توسط بازیکن ارسال نشد، کارگزار باید به انتظار خود خاتمه دهد، بررسی کند که بازی پایان یافته است یا نه و در صورتی که بازی خاتمه نیافته بود، دوباره حداکثر به مدت ۳ ثانیه منتظر حرکت بازیکن بماند. حرکتهای ممکن برای بازیکنها به شکل زیر است:

Action	Action Message
Move Up	UP
Move Down	DOWN
Move Right	RIGHT
Move Left	LEFT

پیغام دریافتی از بازیکنها به یکی از ۴ شکل گفته شده خواهد بود.

امتیازدهی به بازیکن‌ها

نحوه‌ی امتیازدهی به بازیکن‌ها به این شکل است که پس از قرار گرفتن یک بازیکن روی یکی از خانه‌های سکه یا گنج، آن خانه به یک خانه‌ی خالی تبدیل می‌شود و امتیازی معادل با مقادیر جدول زیر به بازیکن اختصاص داده می‌شود:

Type	Score
Coin	10
Gold	100

پایان بازی

بازی زمانی تمام می‌شود که یک بازیکن گنج (خانه‌ی G) را تصاحب کند. پس از اتمام بازی کارگزار با ارسال پیغام زیر به بازیکن‌ها به کار خود خاتمه می‌دهد و بازیکن‌ها هم با گرفتن پیغام، خاتمه پیدا می‌کنند.

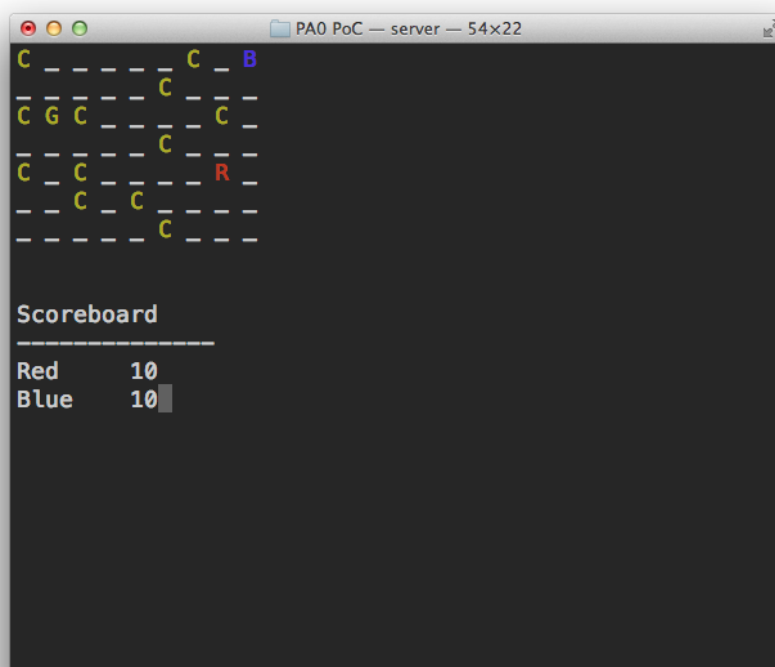
```
<Won-or-Lost>
<final-score-point>
```

برای نمونه، این پیغام به صورت زیر می‌تواند باشد:

```
Won
120
```

نمایش نقشه در برنامه‌ی کارگزار

کارگزار باید در هر لحظه نقشه به همراه امتیاز بازیکن‌ها را نمایش دهد. برای نمایش لحظه به لحظه‌ی خروجی در کنسول باید از کتابخانه‌ی Ncurses استفاده کنید. یک نمونه از نمایش نقشه به شکل زیر است. در این نمایش بهتر است به جای کاراکتر E از Underscore(_) استفاده کنید.



برنامه‌ی بازیکن

این برنامه باید نقش یکی از بازیکن‌های آبی یا قرمز را بازی کند. این برنامه به شکل زیر اجرا می‌شود:

```
$ ./player <server-ip> <server-port> <Red-or-Blue>
```

برای نمونه برای اتصال یک بازیکن آبی به یک کارگزار بازی در آدرس ۱۲۷.۰.۰.۱ روی پورت ۵۰۵۰ دستور زیر را باید وارد کنیم:

```
$ ./player 127.0.0.1 5050 Blue
```

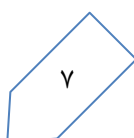
برنامه‌ی بازیکن پس از اتصال موفقیت آمیز به کارگزار باید پیغام زیر را نمایش دهد:

Connected Successfully.

سپس با ارسال پیغام گفته شده در بخش برنامه‌ی کارگزار، قرمز یا آبی بودن خود را به کارگزار اعلام می‌کند. در ادامه کارگزار با ارسال اطلاعات نقشه و امتیاز بازیکن، اطلاعات بازیکن را به‌روز می‌کند و بازیکن هم با گرفتن دستورهای زیر از خط فرمان باید به انجام بازی بپردازد:^۳

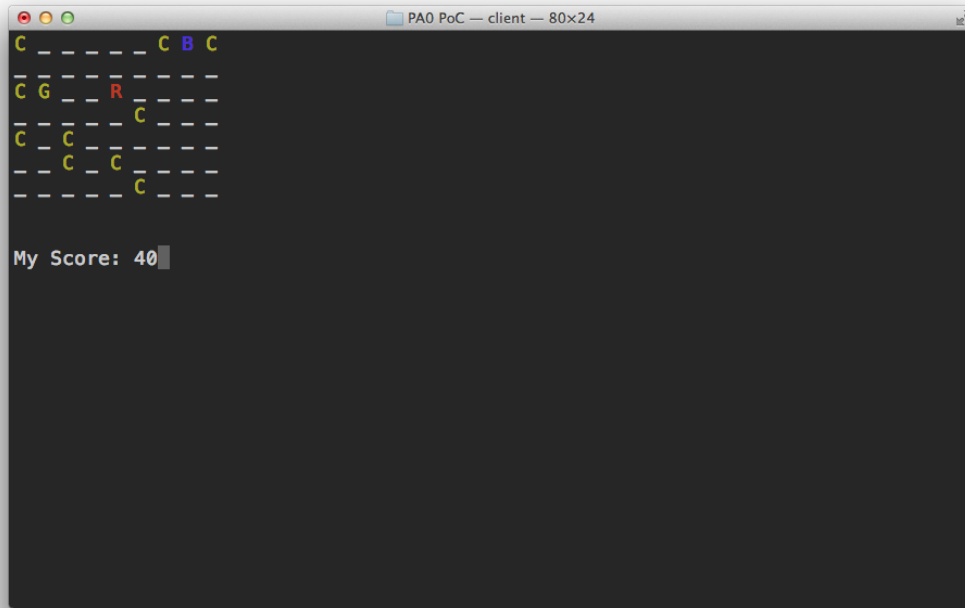
Action	Command-Line Input
Move Up	<Arrow-Up>
Move Down	<Arrow-Down>
Move Right	<Arrow-Right>
Move Left	<Arrow-Left>

^۳ گرفتن کاراکترها باید بدون نیاز به فشردن دگمه‌ی Enter باشد. برای این منظور می‌توانید از تابع `getch()` در کتابخانه‌ی `Ncurses` استفاده کنید.



نمایش نقشه در برنامه‌ی بازیکن

برنامه‌ی بازیکن با پردازش اطلاعات دریافتی از کارگزار، باید نقشه بازی به همراه امتیاز خود را نمایش دهد.

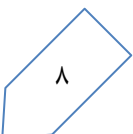


نکات ضروری

- برای آشنایی با نحوه‌ی ساخت thread با کتابخانه‌ی POSIX thread به آدرس زیر مراجعه کنید:
<http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html>
- برای آشنایی با کتابخانه‌ی NCURSES به آدرس زیر مراجعه کنید:
<http://tldp.org/HOWTO/NCURSES-Programing-HOWTO/>
- در صورتیکه هر مشکل یا پرسشی داشتید که فکر می‌کنید پاسخ آن برای همه مفید خواهد بود، لطفاً آن را به گروه پستی درس ارسال کنید.
- از فرستادن جواب تمرین به گروه پستی جداً خودداری کنید.
- فرستادن کل یا قسمتی از برنامه‌تان برای افراد دیگر، یا استفاده از کل یا قسمتی از برنامه‌ی فردی دیگر به نام خود، تقلب محسوب می‌شود.
- پس از اتمام کارتان لازم است که فایل‌های خود را به همراه Makefile فشرده کرده و بر روی سیستم خودکار داوری⁴ upload کنید.

موفق باشید

⁴ <http://partov.sharif.edu/judge>



ضمیمه: آشنایی با کتابخانه‌ی Ncurses

کتابخانه Ncurses این امکان را فراهم می‌کند که با کنسول به شکل یک پنجره رفتار کنید. این کتابخانه با استفاده از امکانات موجود در محیط کنسول، امکاناتی مانند نوشتن رنگی و... را در این محیط با توابع سطح بالای خود فراهم می‌کند. در ادامه چند تابع پر کاربرد این کتابخانه آورده شده است:

نام تابع	عملکرد
initscr	رفتن به حالت نمایش Ncurses
printw	تابعی مشابه printf که در حالت نمایش Ncurses کار می‌کند.
mvprintw	تابعی مشابه printw که در یک مختصات خاص متن موردنظر را نمایش می‌دهد.
mvaddch	تابعی برای نمایش یک کاراکتر در یک مختصات خاص
start_color	تابعی برای شروع امکان نمایش رنگی کاراکترها. برای دیدن نمونه به اینجا مراجعه کنید.
clear	تابعی برای پاک کردن کل صفحه‌ی نمایش
refresh	بعد از صدا زدن هر یک از توابع کشیدن، باید با صدا زدن این تابع صفحه را به‌روز کنید تا تغییرات اعمال شده را نمایش دهد.
getch	این تابع برای گرفتن یک کاراکتر استفاده می‌شود. برخلاف توابع استاندارد زبان C و C++ این تابع نیاز به زدن Enter برای گرفتن ورودی ندارد. با کمک تابع keypad در همین کتابخانه می‌توانید امکان گرفتن کاراکترهایی مانند Arrow Key ها را نیز فعال کنید. همچنین از این تابع برای ایجاد pause در برنامه‌ی خود می‌توانید استفاده کنید.
endwin	پس از پایان برنامه باید این تابع صدا زده شود تا کنسول به حالت عادی خود بازگردد و از حالت Ncurses خارج شود.