

برنامه‌سازی پیشرفته

تمرین کامپیوتری شماره ۶

مدرس: رامتین خسروی



طراحان: محمدعرفان دانایی، پارسا دقیق، امیرعلی دهقانی، امیرحسین عارفزاده، بهاره عین‌الهی، عرفان فلاحتی، طاها مجلسی، مهدیس میرزائی، امیررضا نادی، پریسا یحیی‌پور

مهلت تحویل: چهارشنبه ۱۲ خرداد ۱۴۰۴، ساعت ۲۳:۵۹



مقدمه

این پروژه به جمع‌بندی آموخته‌های شما در این درس می‌پردازد. انتظار می‌رود مهارت‌هایی را که در تمرین‌های پیشین و سایر بخش‌های درس آموخته‌اید، در پیاده‌سازی این پروژه به کار گیرید. در این فاز پروژه، تعدادی دستور در راستای گسترش امکانات برنامه به آن افزوده و پیاده‌سازی می‌شوند. قالب این دستورات مانند فاز قبلی می‌باشد و می‌توانید برای بررسی دقیق‌تر به فاز قبل مراجعه کنید. همچنین باید تمام دستورات فاز قبلی در این فاز نیز قابل اجرا باشد (به جز دستوراتی که در این فاز دستخوش تغییر شده‌اند، این دستورات باید به صورت جدید اجرا شوند). قابلیت جدیدی که در این فاز به آن می‌پردازیم، رویداد مشترک است. رویدادی مانند جلسات آنلاین در بستر گوگل‌میت که بین چند نفر به شکل مشترک برگزار می‌شود.

دستورات جدید

اضافه کردن رویداد مشترک

یکی از قابلیت‌های برنامه این است که چند شخص می‌توانند با یکدیگر رویداد مشترک داشته باشند. هر رویداد مشترک شامل یک میزبان و یک یا چند مهمان است. شخصی که این دستور را وارد می‌کند، میزبان رویداد مشترک است. توجه داشته باشید تا زمانی که مهمان رویداد مشترک را تایید نکند، به برنامه زمانی او اضافه نمی‌شود. همچنین رویداد مشترک در صورتی به برنامه زمانی میزبان اضافه می‌شود که حداقل یکی از مهمان‌ها، رویداد مشترک را تایید کند.

رویدادهای مشترک شناسه دارند و این شناسه به طور خودکار توسط سیستم اختصاص داده می‌شود. این شناسه برای کل برنامه از ۱ شروع می‌شود و یکی یکی افزایش می‌یابد¹.

دقت کنید، به رویداد معمولی و دوره‌ای هر کاربر مستقل از کاربران دیگر شناسه تعلق می‌گیرد. اما چون رویداد مشترک فقط مختص به یک کاربر نیست، شناسه آن در کل برنامه تعیین می‌شود. برای مثال اگر کاربر شماره ۱ و سپس کاربر شماره ۲ به ترتیب رویداد مشترک ایجاد کنند، شناسه رویداد مشترک به میزبانی کاربر شماره ۱ برابر ۱ و شناسه رویداد مشترک به میزبانی کاربر شماره ۲ برابر ۲ خواهد بود.

اگر رویداد مشترک چند مهمان داشته باشد، نام کاربری مهمان‌ها با استفاده از کاما (,) از هم جدا می‌شوند نام کاربری هر مهمان بین دو کاراکتر " قرار می‌گیرد.

همچنین اگر این رویداد با رویدادهای معمولی، دوره‌ای یا مشترک دیگر در برنامه زمانی میزبان تداخل داشته باشد، پیام Overlap چاپ می‌شود و دعوت‌نامه رویداد برای هیچ‌کس ارسال نمی‌شود.

در صورتی که رویداد مشترک در تعطیلات رسمی باشد، پیام Holiday Found چاپ می‌شود و دعوت‌نامه رویداد برای هیچ‌کس ارسال نمی‌شود. توجه کنید بین تداخل (Overlap) و تعطیلات رسمی (Holiday Found) اولویت با تداخل است.

به جز بخش description، تمام ورودی‌ها برای این دستور الزامی هستند و ورودی guests باید حداقل یک ورودی داشته باشد.

مقدار بخش description و title بین دو کاراکتر " قرار می‌گیرند.

تضمین می‌شود میزبان نام کاربری خود را به عنوان مهمان وارد نمی‌کند. دعوت‌نامه رویداد برای خود میزبان ارسال نمی‌شود.

¹ برای توضیحات بیشتر به این بخش مراجعه کنید

قالب ورودی
<pre>POST join_event ? guests "<guest_username1>,<guest_username2>,..." date <YYYY/MM/DD> start_time <HH> end_time <HH> title "<title>" description "<description>"</pre>
قالب خروجی
Overlap Holiday Found OK Permission Denied Bad Request Not Found

خطاها

- در صورتی که اجزای دستور به درستی وارد نشوند، با خطای Bad Request مواجه می‌شویم.
- اگر نام کاربری حداقل یکی از مهمان‌ها وجود نداشته باشد، با خطای Not Found مواجه می‌شویم.
- در صورتی که هیچ کاربری login نکرده باشد، با وارد کردن این دستور خطای Permission Denied چاپ می‌شود.
- در صورتی که رویداد مشترک با سایر رویدادها (مشترک، معمولی یا دوره‌ای) تداخل داشته باشد، خطای Overlap چاپ می‌شود.

نمونه ورودی ۱
<pre>POST join_event ? guests "ali" date 1404/04/04 start_time 12 end_time 13 title "Hello" description "World"</pre>
نمونه خروجی ۱
OK

نمونه ورودی ۲
<pre>POST join_event ? guests "parisa,erfan" date 1405/05/05 start_time 7 end_time 10 title "Meow"</pre>
نمونه خروجی ۲
OK

مشاهده دعوت‌نامه‌ها

به کمک این دستور کاربر می‌تواند لیست دعوت‌نامه‌هایی که برای او ارسال شده‌اند را مشاهده کند و به کمک دستورهای بعدی، این دعوت‌نامه‌ها را رد یا تایید کند. لیست دعوت‌نامه را به شکل صعودی بر اساس شناسه آنها چاپ کنید.

قالب ورودی
<code>GET join_event ?</code>
قالب خروجی
<code><join_event_id1>: "<title1>" - <date1> - <start_time1> - <end_time1> <join_event_id2>: "<title2>" - <date2> - <start_time2> - <end_time2> . . . Empty Permission Denied Bad Request</code>

خطاها

- در صورتی که اجزای دستور به درستی وارد نشوند، با خطای Bad Request مواجه می‌شویم.
- اگر کاربر هیچ دعوت‌نامه‌ای نداشته باشد، خطای Empty چاپ می‌شود.
- در صورتی که هیچ کاربری login نکرده باشد، با وارد کردن این دستور خطای Permission Denied چاپ می‌شود.

نمونه ورودی
<code>GET join_event ?</code>
نمونه خروجی
<code>1: "Hello" - 1404/04/04 - 12 - 13 2: "Hi" - 1404/03/03 - 7 - 8</code>

تایید رویداد مشترک

کاربر به کمک این دستور رویداد مشترکی را تایید می‌کند. در این صورت این رویداد به برنامه زمانی او اضافه می‌شود. قبل از اضافه شدن این رویداد مشترک به برنامه زمانی، تداخل آن با سایر رویدادها بررسی می‌شود و اگر این رویداد با رویدادهای معمولی، دوره‌ای یا مشترک در برنامه زمانی کاربر تداخل داشته باشد، پیام Overlap چاپ می‌شود و رویداد به برنامه زمانی کاربر اضافه نمی‌شود. توجه کنید بعد از وارد کردن این دستور، این رویداد مشترک از لیست دعوت‌نامه‌های کاربر حذف می‌شود. ورودی invitation_id در این دستور الزامی است.

قالب ورودی
<code>POST confirm_join_event ? invitation_id <join_event_id></code>
قالب خروجی
Overlap OK Bad Request Not Found Permission Denied

خطاها

- در صورتی که اجزای دستور به درستی وارد نشوند، با خطای Bad Request مواجه می‌شویم.
- اگر رویداد مشترکی با شناسه وارد شده در بین دعوت‌نامه‌های کاربر وجود نداشته باشد، با خطای Not Found مواجه می‌شویم.
- در صورتی که هیچ کاربری login نکرده باشد، با وارد کردن این دستور خطای Permission Denied چاپ می‌شود.

نمونه ورودی ۱
<code>POST confirm_join_event ? invitation_id 1</code>
نمونه خروجی ۱
OK

نمونه ورودی ۲
<code>POST confirm_join_event ? invitation_id 2</code>
نمونه خروجی ۲
Overlap

رد رویداد مشترک

کاربر با وارد کردن این دستور، رویداد مشترکی را رد می‌کند و این رویداد از لیست دعوت‌نامه‌های کاربر حذف می‌شود. ورودی invitation_id در این دستور الزامی است.

قالب ورودی
<code>POST reject_join_event ? invitation_id <join_event_id></code>
قالب خروجی
<code>OK Bad Request Not Found Permission Denied</code>

خطاها

- در صورتی که اجزای دستور به درستی وارد نشوند، با خطای Bad Request مواجه می‌شویم.
- اگر رویداد مشترکی با شناسه وارد شده در بین دعوت‌نامه‌های کاربر وجود نداشته باشد، با خطای Not Found مواجه می‌شویم.
- در صورتی که هیچ کاربری login نکرده باشد، با وارد کردن این دستور خطای Permission Denied چاپ می‌شود.

نمونه ورودی
<code>POST reject_join_event ? invitation_id 1</code>
نمونه خروجی
<code>OK</code>

گزارش

این دستور همان دستور فاز قبلی است. اما با توجه به اضافه شدن رویداد مشترک به برنامه در این فاز، گزارش‌گیری از رویداد مشترک به آن اضافه شده‌است.

با استفاده از این دستور، لیستی از تمام کارهایی که کاربر در برنامه زمانی خود دارد، نمایش داده می‌شود. ترتیب نمایش خروجی بر حسب نزدیک‌ترین فعالیت است. در صورتی که انتخاب بیش از یک کار وجود داشت، بر اساس نوع فعالیت به ترتیب رویداد مشترک، رویداد دوره‌ای، رویداد معمولی و تسک اولویت‌بندی کنید. در صورتی که باز هم بیش از یک انتخاب وجود داشته باشد، بر اساس شناسه به شکل صعودی اولویت بدهید.

آرگومان to برای این دستور اجباری است اما آرگومان‌های type و from اختیاری هستند. اگر آرگومان from وارد نشود، باید تمامی کارها از تاریخ مبدا سامانه تا تاریخ مشخص شده چاپ شود. همچنین این دستور می‌تواند شامل یک آرگومان اختیاری type باشد که فقط باید کارهایی از آن نوع (رویداد معمولی، رویداد دوره‌ای، رویداد مشترک یا تسک) را چاپ کند. مقادیر مجاز برای آرگومان type شامل موارد زیر است:

- رویداد معمولی: event
- رویداد دوره‌ای: periodic_event
- رویداد مشترک: join_event
- تسک: task

تضمین می‌شود مقادیر آرگومان type در صورت وجود یکی از این چهار کلمه است. اگر آرگومان type وارد نشود، تمام کارها را خروجی دهید.

قالب ورودی
GET report ? from <YYYY/MM/DD> to <YYYY/MM/DD> type <type>
قالب خروجی
task "<title>" on <YYYY/MM/DD> at <time>: "<description>" event "<title>" on <YYYY/MM/DD> from <start_time> for <duration> hours: "description" periodic_event "<title>" on <YYYY/MM/DD> from <start_time> for <duration> hours <type>: "<description>" join_event "<title>" on <YYYY/MM/DD> from <start_time> to <end_time> hosted by "<host_username>" : "<description>" Permission Denied Bad Request Empty

خطاها

- اگر کاری وجود نداشت باید خطای Empty چاپ می‌شود.
- در صورتی که هیچ کاربری login نکرده باشد، با وارد کردن این دستور خطای Permission Denied چاپ می‌شود.
- در صورتی که اجزای دستور به درستی وارد نشوند، با خطای Bad Request مواجه می‌شویم.

نمونه ورودی ۱
<code>GET report ? from 1404/01/23 to 1404/01/29</code>
نمونه خروجی ۱
<pre>periodic_event "AP Class" on 1404/01/24 from 10 for 1 hours Weekly: "Advanced Programming course with Dr. Khosravi" task "Assignment 3" on 1404/01/25 at 23: "This assignment is about Backtracking and Recursive Functions" periodic_event "AP Class" on 1404/01/26 from 10 for 1 hours Weekly: "Advanced Programming course with Dr. Khosravi" event "Meeting with Mahdis" on 1404/01/29 from 16 for 1 hours: join_event "Boo" on 1404/01/29 from 17 to 18 hosted by "parisa": "Doo"</pre>

نمونه ورودی ۲
<code>GET report ? to 1404/01/02 type task</code>
نمونه خروجی ۲
<code>Empty</code>

ساخت شناسه

در فاز قبل تمام موجودیت‌های سامانه، شناسه در سطح کاربر داشتند. در این فاز رویداد مشترک موجودیتی است که شناسه آن در سطح برنامه تولید می‌شود.

سطح برنامه

در این سطح شناسه اختصاص داده شده به موجود، در کل برنامه یکتا است. به این معنا که در ابتدای اجرای برنامه از ۱ شروع می‌شود و یکی یکی به شکل صعودی اضافه می‌شود.

مثال

1. کاربر شماره ۱ یک رویداد مشترک ایجاد می‌کند. شناسه آن برابر ۱ است.
2. کاربر شماره ۲ یک رویداد مشترک ایجاد می‌کند. شناسه آن برابر ۲ است.
3. کاربر شماره ۱ رویداد مشترک دیگر ایجاد می‌کند. شناسه آن برابر ۳ است.

نکات و نحوه تحویل

- برای تحویل این پروژه، لازم به ساخت مخزن² جدید نیست و باید کد این فاز را در همان مخزن فاز اول کامیت³ کنید. در صورتی که برای فاز اول مخزن ایجاد نکردید، لطفاً از طریق [این لینک](#) وارد شوید و شماره دانشجویی خود را انتخاب کنید (دقت کنید که با کمک این شماره دانشجویی به شما نمره خواهیم داد، لطفاً در انتخاب درست شماره دانشجویی حتماً دقت کنید، در صورتی که به مشکل خوردید با دستیاران در ارتباط باشید).
- پس از انجام تمرین و بارگذاری در گیت‌هاب، کد Hash آخرین کامیت⁴ را به همراه شماره دانشجویی خود در سامانه ایلرن آپلود کنید (در خط اول شماره دانشجویی، پس از آن از Enter استفاده کنید و به خط بعد بروید و پس از آن Hash آخرین کامیت). نمونه متن خواسته شده در سامانه ایلرن بخش <last_commit_hash> و <sid> را جایگزین کنید):

```
<sid>  
<last_commit_hash>
```

نمونه:

```
810100000  
bad8fbcdcf3b9feb371a31e0c370150aa870b18
```

- دقت کنید که عدم رعایت ساختار گفته شده در آپلود یا تغییر ساختار فایل‌ها در مخزن (می‌توانید به دلخواه خود فایل اضافه کنید و ... اما اسم و ساختار فایل‌هایی که در ابتدا به شما داده می‌شود نباید تغییر کند) باعث کسر 5 درصد از نمره شما خواهد شد.
- دقت کنید که فایل makefile باید در صفحه اول مخزن باشد و در پوشه‌ای قرار نداشته باشد و در آن مشخص کنید که از استاندارد c++20 استفاده می‌کنید.
- نام برنامه قابل اجرای شما باید UTrello (بدون هیچ پسوندی مانند exe یا out) باشد و پس از ساخته شدن در کنار makefile قرار بگیرد (داخل پوشه‌ای فایل خروجی ساخته شده را قرار ندهید).
- درستی برنامه شما از طریق آزمون‌های خودکار سنجیده می‌شود؛ بنابراین پیشنهاد می‌شود که با استفاده از ابزارهایی مانند diff خروجی برنامه خود را با خروجی‌هایی که در اختیارتان قرار داده شده است مطابقت دهید.
- برنامه شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++20 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.

² Repository

³ Commit

⁴ Commit

- درستی برنامه شما از طریق آزمون‌های خودکار سنجیده می‌شود؛ بنابراین از درستی کامل قالب خروجی برنامه خود اطمینان حاصل کنید و از دادن خروجی‌هایی که در صورت پروژه ذکر نشده است اجتناب کنید.
- سوالات خود را تا حد ممکن در فروم درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آن‌ها بهره‌مند شوند. در صورتی که قصد مطرح کردن سوال خاص‌تری داشتید، از طریق ایمیل با طراحان این فاز پروژه ارتباط برقرار کنید.
- توجه داشته باشید که حالت‌های خاصی که در صورت پروژه ذکر نشده است در تست‌های خودکار نخواهد بود و می‌توانید به هر شکلی که مد نظر دارید آن‌ها را مدیریت کنید.
- هدف این تمرین یادگیری شماست. لطفا تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق سیاست درس با آن برخورد خواهد شد.

نمرات

- تمیزی کد
 - رعایت کردن نام‌گذاری صحیح و انسجام⁵
 - عدم وجود کد تکراری
 - رعایت دندانه‌گذاری⁶
 - عدم استفاده از متغیرهای گلوبال
 - استفاده صحیح از متغیرهای ثابت⁷ به جای Magic Value-ها
 - ساختاردهی کد در قالب توابع کوتاه که فقط یک کار را انجام می‌دهند
- درستی کد
 - آزمون‌های خودکار
 - پیاده‌سازی صحیح کارکردهای خواسته شده
- طراحی
 - طراحی صحیح و منطقی در شی‌گرایی و ارث‌بری
 - رعایت Encapsulation
 - جداسازی منطق کد از ورودی/خروجی و استفاده از کلاس جداگانه برای مدیریت دستورات
 - استفاده مناسب از استثناها برای مدیریت خطا
 - میک‌فایل و چندفایلی

دقت کنید که موارد ذکر شده لزوماً کل نمره شما را تشکیل نمی‌دهند و ممکن است با تغییراتی همراه باشند.

⁵ Consistency

⁶ Indentation

⁷ Constant