به نام خدا

درس مبانی برنامه‌سازی

فاز دوم پروژه

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نیم سال اول ۰۲ ـ ۰۱

استاد:

دکتر محمدامین فضلی

مهلت ارسال:

فاز دوم: ۱۲ بهمن
ساعت ۲۳:۵۹:۵۹

مسئول پروژه:

امیرمهدی کوششی

مسئول فاز دوم:

آرمان بابائی

طراحان فاز دوم:

محمدمهدی قیدی، سروش شرافت، عرفان مجیبی، زهرا رحمانی، محمد ایزدی، محمد خلفی

مسئولین تنظیم مستند:

امیرمهدی کوششی، آرمان بابایی

فهرست

نکات قابل توجه ۲

مقدمه ۳
اهداف قابل توجه ۳
کلیات پروژه ۳

معرفی ابزار ۴

توضیح بخش‌های مختلف پروژه ۵
موارد عمومی و قابلیت‌های ابزار ۵
Window Layout ۵
Navigation ۶
Selection ۷
Clipboard ۷
Saving ۸
Open ۸
Undo ۸
Find ۹
Replace ۹
Auto-indent ۹
دستورات فاز اول ۱۰

نکات قابل توجه

- پس از اتمام این فاز، در گیت خود یک تگ با ورژن "v2.0.0" بزنید. در روز تحویل حضوری این tag بررسی خواهد شد و کدهای پس از آن نمره‌ای نخواهد گرفت. برای اطلاعات بیش‌تر در مورد شیوه ورژن‌گذاری، می‌توانید به این لینک مراجعه کنید. البته برای این پروژه صرفا رعایت کردن همان ورژن گفته شده کافیست، اما خوب است که با منطق ورژن‌بندی هم آشنا بشوید.
- در صورت کشف تقلب، برای بار اول منفی نمرهٔ آن فاز برای آن فرد ثبت می‌شود و برای بار دوم، نمرهٔ منفی کل پروژه برای فرد لحاظ خواهد شد.

مقدمه

اهداف پروژه

- هدف این پروژه طراحی یک ابزار ویرایش فایل مشابه vim است. احتمالا در کارگاه کامپیوتر و یا جاهای دیگر با این ابزار کار کرده‌اید. در غیر این صورت می‌توانید از طریق این لینک نحوه کار با این ابزار را ببینید.
- در این فاز از برای ابزاری که در فاز قبل طراحی کردید رابط کاربری می‌سازید و برخی ویژگی‌ها را بهبود می‌دهید.
- در این پروژه نحوه پیاده‌سازی اجزای مختلف از اهمیت بسیاری برخوردار است و تنها خروجی نهایی مهم نیست. از این رو برای تمیزی کد خود ارزش قائل شوید.
- آشنایی با سیستم مدیریت نسخه Git و کار بر روی پروژه بر بستر یک مخزن Github، یکی از اهداف مهم پروژه است. در این مورد توصیه می‌شود تغییرات خود را در دوره‌های کوتاه مدت commit کنید.

کلیات پروژه

در این فاز، کد فاز اول خود را کامل می‌کنید.
در ادامهٔ مستند، موجودیت‌ها، نمای کلی رابط کاربری سیستم، نقش‌ها و دستورات لازم شرح داده‌شده است.

نکته ۱: در هر جایی از پروژه می‌توانید هرگونه خلاقیتی را به‌کار ببرید. با این حال توجه کنید که خواسته‌های واضح پروژه بایستی انجام شوند و سیستم ورودی گرفتن و خروجی دادن شما باید مطابق جزییات گفته شده در این مستند باشد.

نکته ۲: در این فاز برخی از دستورات را با استفاده کلیدهای شورتکات پیاده‌سازی می‌کنید. کلیدهای ذکرشده در ادامه‌ی مستند پیشنهادی هستند و می‌توانید به صلاح‌دید خود آن‌ها را تغییر دهید. توجه کنید که این کار نباید منجر به محدود شدن کاربری برنامه‌ی شما شود.

معرفی ابزار

همانطور که قبلا توضیح دادیم، شما باید یک ابزار ویرایش متن مانند vim طراحی کنید. در این فاز از پروژه، شما باید یک محیط گرافیکی مانند خود vim بالا بیاورید و تمامی دستوراتی که به برنامه خود می‌دهید از طریق همین محیط گرافیکی است. شما باید از طریق محیط گرافیکی دستورات را بگیرید و پردازش کنید و تغییرات را روی فایل‌ها اعمال کنید و در نهایت خروجی مناسب را به صورت زنده (live) به کاربر نمایش دهید.

توضیح بخش‌های مختلف پروژه

موارد عمومی و قابلیت‌های ابزار

در این فاز بنابر این داریم تا رابط گرافیکی یا GUI مربوط به فاز اول را پیاده‌سازی نماییم. در فاز اول مواردی پیاده‌سازی کردید که با ترمینال انجام دادید و در این فاز موارد گرافیکی مربوط به vim را خواهیم نوشت.

Window Layout

برای نشان دادن مواردی که کاربر تایپ می‌کند و محتویات فایل مورد بررسی نیاز داریم تا یک پنجره داشته باشیم تا پایه‌ریزی کلی یا layout مواردی که قرار است نشان داده شوند را در آن به نمایش بگذاریم. یک پنجره vim به طور کلی باید قادر به نمایش ۵ بخش زیر باشد که به همراه تطابق آن‌ها با تصویر آمده‌اند:

- اسم فایل و وضعیت ذخیره شدن آن در دیسک (ذخیره شده/نشده)
  اسم فایل روبروی کلمه‌ی NORMAL آمده است. ذخیره‌نشده بودن فایل با استفاده از علامت + در کنار اسم فایل نشان داده شده است.
- حالت فعلی vim یا mode
  کلمه‌ی NORMAL در تصویر به این کار اختصاص داده شده.
- خط دستور یا bar command که دستوراتی که در ادامه داک می‌آیند در این قسمت وارد می‌شوند.
  این قسمت در آخرین خط صفحه آمده است. با زدن کلید / و یا : در حالت NORMAL کاربر شروع به تایپ در این محدوده می‌کند و تا زمانی که کلید enter را فشار نداده به این کار ادامه می‌دهد.
- شماره هر خط فایل
  این شماره‌ها در سمت چپ خطوط نوشته شده‌اند.
- و در نهایت محتویات فایل

ادیتور vim به صورت کلی ۳ حالت اصلی insert و normal و visual را دارد. در هر کدام از این حالت‌ها کارهای بخصوصی می‌توان انجام داد که در ادامه داک و برای هر دستور ذکر شده است که باید در کدام حالت اعمال شوند. برای گرفتن شهود بیشتر میتوانید عکس زیر را مشاهده کنید که می‌تواند پیاده‌سازی مطلوبی از موارد بالا باشد.

```
  1 function factorial(n) {
  2     let result = 1;
  3     for (let i = 1; i <= n; i++) {
  4         result *= i;
  5     }
  6     return result;
  7 }
  8 console.log(factorial(5));
  9
~
~
~
~
~
 NORMAL   factorial   +
:command args
```

Navigation

برنامه‌ی شما باید قابلیت حرکت دادن نشانگر (cursor) با استفاده از کیبورد بر روی متون را داشته باشد. مواردی که باید پیاده سازی به آنها توجه کنید:

- کاربر باید بتواند نشانگر (cursor) را بر روی بخشی از متن که در حال نمایش است به چپ و راست و بالا و پایین حرکت دهد. (کلیدهای پیشنهادی: بالا(k) پایین(j) چپ(h) و راست(l))
- با رسیدن به ابتدای هر خط، با فشردن کلید چپ و همچنین با رسیدن به انتها خط و فشردن کلید راست، نباید اتفاقی بیفتد.
- انتقال به خط بعد/قبل (بالا و پایین رفتن) باید به مکان نسبی مشابه مکان نسبی نشانگر (cursor) در خط فعلی باشد. (برای مثال اگر در خط فعلی در کاراکتر بیستم قرار داریم، با فشردن کلید j باید در خط بعد و کاراکتر بیستم قرار داشته باشیم.) اگر تعداد کاراکتر خط بعدی کمتر بود، به آخرین کاراکتر آن انتقال یابیم.
- در ۴ خط مانده به پایان صفحه فعلی، با فشردن کلید حرکت به پایین، باید یک خط شیفت دهید. یعنی یک خط جدید از فایل به صفحه اضافه شده و خط اول حذف شود. همچنین وقتی نشانگر (cursor) در محل ۴ خط مانده به ابتدای صفحه فعلی است، در صورت فشردن کلید بالا، باید یک خط از بالای صفحه اضافه شده و یک خط از پایین حذف شود.

- لازم است توجه کنید با کلید پایین رفتن باید به خط بعدی با توجه کاراکتر \n بروید و شکل ظاهری خطوط ملاک نیست.

Selection

در حالت بصری یا همان visual امکان این را داریم که قسمتی از محتویات فایل را انتخاب (select) کرده و با آن قسمت انتخاب شده مواردی چون copy یا cut یا delete را بتوانیم انجام دهیم. در پیاده‌سازی شما از محل نشانگر (cursor) در زمان ورود به حالت visual تا محل فعلی نشانگر (cursor) باید انتخاب شود. می‌توانید این انتخاب را با هایلایت کردن قسمت انتخاب‌شده نشان دهید.

توجه کنید که در حالت بصری همچنان قابلیت جابجا کردن نشانگر (cursor) وجود دارد و این حرکت می‌تواند به سمت بعد یا قبل (و یا ترکیبی از این دو) محل نشانگرa (cursor) در هنگام ورود به حالت بصری باشد.

Clipboard

پس از انتخاب قسمتی از متن در حالت visual باید این امکان برای کاربر وجود داشته باشد که متن انتخاب شده را حذف، قیچی و یا روگرفت بکند. کاربر باید بتواند در حالتی که متن انتخاب شده، به وسیله هر کدام از دستورات زیر عملیات مورد نظر خودش را انجام بدهد.

- Cut/Delete: d
- Copy: y

بدین منظور باید یک فیچر clipboard برای ادیتور خود پیاده‌سازی کنید تا قسمت‌هایی که cut/copy می‌شوند در آن قرار بگیرند. پس از هر کدام از کلیدهای مربوط به روگرفت و یا قیچی باید از حالت visual به حالت normal منتقل شوید. توجه کنید که دستورات قیچی و روگرفت فاز اول هم در همین دسته محسوب می‌شوند و فرقی نمی‌کند که از کلیدها و یا دستورات فاز اول برای اضافه کردن متن به clipboard استفاده شده باشد.

دستور paste که در حالت normal تعریف می‌شود به کاربر این امکان را می‌دهد که در این حالت بتواند به وسیله کلید p مقدار ذخیره‌شده در clipboard برنامه (که حاصل اجرای یک دستور cut یا copy بوده است) را در محل نشانگر (cursor) قرار دهد.

Saving

برنامه شما باید به کاربر این امکان را بدهد که با زدن کامند

دستور

```
:save
```

فایل فعلی خود را ذخیره کند. در صورتی که کاربر قبلاً برای فایل خود اسم انتخاب نکرده باشد باید با چاپ پیام مناسب (احتمالا در محل ورودی گرفتن دستور) از او بخواهید برای فایل خود اسم انتخاب کند. همچنین با زدن کامند

دستور

```
:saveas <name>
```

کاربر باید بتواند فایل را با نام دلخواهش ذخیره کند. پس از تکمیل فرایند ذخیره نیز با نمایش پیام مناسب موفقیت‌آمیز بودن ذخیره را به کاربر اطلاع دهید.

Open

در این فاز باید امکان باز کردن یک فایل جدید در برنامه را پیاده سازی کنید. با وارد کردن این کامند باید یک فایل جدید در ویرایشگر باز شود. اگر از قبل فایلی باز بوده، محتوای آن باید سیو شود و بسته شود و فایل جدید جایگزین آن شود.

دستور

```
:open <file>
```

Undo

این امکان را در فاز قبل پیاده سازی کرده‌اید و باید یک شورتکات هم برای آن تعریف کنید. شورتکات پیشنهادی کلید u در حالت نرمال است. بنابراین کاربر حداقل به دو صورت امکان undo داشته باشد:

- زدن دکمه u در حالت نرمال
- زدن کامند undo:

Find

با وارد کردن کامند

دستور

/<expression>

باید همه‌ی رخدادهای <expression> را در متن پیدا و آن ها highlight کنید. همچنان باید این قابلیت را پیاده سازی کنید که با استفاده از کلید n، نشانگر (cur-sor) به اولین مکان رخداد <expression> پس از مکان فعلی نشانگر (cursor) برود. (اگر چنین رخدادی وجود نداشت نشانگر (cursor) باید به اولین رخداد برود و اگر هیچ رخدادی وجود نداشت مکان نشانگر (cursor) نباید تغییر کند) دقت کنید که پس وارد کردن هر کلید یا کامندی به جز کلید n باید هایلایت‌های مربوط به کامند find را پاک کنید.

Replace

دستور replace با تغییر اندک مشابه فاز اول باقی می‌ماند. در صورتی که نام فایل در دستور آمده بود (منظور همان آرگومان file- است) با این دستور مشابه دستورات دیگر فاز اول برخورد کنید، اما در صورتی که آرگومان file- در دستور وجود نداشت عملیات را در فایلی که در ویرایشگر باز است انجام دهید. علاوه بر این در صورتی که تغییری در فایل ایجاد شد، نشانگر (cursor) را به موقعیت شروع اولین تغییر انتقال دهید.

مثال

:replace –str1 ”salam khubi?” –str2 ”Dorud! Che Khabar?” -at 2

در نتیجه‌ی این دستور باید دومین مقدار رشته‌ی str۱ در فایل با str۲ جابجا شود و نشانگر (cursor) به جایی که دومین str۱ در آن قرار داشت، منتقل شود.

Auto-indent

این دستور را در فاز یک پیاده سازی کرده‌اید. در این فاز باید علاوه بر در نظر گرفتن کامند auto-indent برای اجرای این دستور، یک shortcut نیز برای آن تعریف کنید. کلید پیشنهادی برای این منظور کلید = است. واضح است که تعریف این کلید در حالت normal کافی‌ست.

دستورات فاز اول

این فاز هیچ کدام از دستورات فاز اول را منسوخ نمی‌کند؛ به این معنی که تمام دستورهایی که در فاز قبل وجود داشتند مثل insertstr یا arman یا... در این فاز هم به همان شکل وجود دارند. حتی دستوری مثل find که در این فاز به نوع جدیدی مطرح شده، باید به صورت فاز اول هم قابل‌استفاده باشد. تنها تغییری که در این دستورات به وجود می‌آید این است که یک کاراکتر ":" به ابتدای این دستورات اضافه می‌شود.

مثال

```
:createfile file_name.txt
```

خروجی این دستورات (در صورت وجود) باید به صورت یک فایل بدون نام در ویرایشگر باز شوند. پس مثلا اگر دستور

مثال

```
:tree
```

وارد شد، لازم است فایل فعلی در صورت وجود بسته شود، و یک فایل بدون نام با محتوای درخت مشابه فاز اول باز شود. واضح است که امکان جستجو، تغییر و ذخیره‌سازی مثل هر فایل دیگری برای این فایل هم تعریف شده است. با این وجود در صورتی که این فایل بدون ذخیره‌سازی بسته شد (یا فایل جدیدی باز شد) لازم است هیچ اثری از این فایل باقی نماند.