

پروژه پایانی

TUI To-Do List

دکتر ملکی مجد

۱۰ دی ۱۴۰۳

طراحان

میلاد زارعی ملکی

سعید نوریان

مهدی قضاوی

مرتضی جوادیان



دانشگاه علم و صنعت ایران

پاییز ۴۰۳۱

۱ نکات پروژه

- تنها زبان مورد قبول برای پروژه C است.
- پروژه به صورت تک نفره انجام می‌شود.
- ددلاین پروژه قابل تمدید نیست، لذا در مدیریت زمان خود دقت کنید.
- پروژه بایستی پس از اتمام به تی ای مربوطه ارائه داده شود.
- بدون ارائه پروژه، نمره صفر خواهد شد.

۲ طراحان پروژه

در صورت داشتن هر گونه سوال یا مشکل در مورد پروژه، می‌توانید با طراحان پروژه در ارتباط باشید یا در گروه تلگرامی، سوال خود را مطرح کنید.

• میلاد زارعی ملکی - @LibreArtisan

• سعید نوریان - @Saeedvft

• مهدی قضاوی - @iMahdiGhazavi

• مرتضی جوادیان - @mor1383teza

۳ معرفی پروژه

پروژه حاضر با هدف طراحی یک رابط کاربری متنی (TUI یا Text-based User Interface) توسعه داده شده است که می‌تواند تجربه‌ای جذاب و حرفه‌ای از کار با ابزارهای متنی در ترمینال ارائه دهد. این پروژه این امکان را فراهم می‌کند تا مهارت‌های برنامه‌نویسی خود را در زبان C با بهره‌گیری از کتابخانه NCurses تقویت کنید.

در این پروژه شما موظف هستید که یک برنامه مدیریت تسک‌ها را در فضای TUI پیاده‌سازی کنید که به کاربران قابلیت‌های متعددی می‌دهد تا بتوانند تسک‌های خود را به راحتی مدیریت کنند.

۱.۳ رابط کاربری متنی (TUI)

رابط کاربری متنی یا TUI نوعی واسط کاربری است که به جای استفاده از المان‌های گرافیکی (مانند دکمه‌ها و پنجره‌های معمول)، از متن و کاراکترها برای نمایش و تعامل با کاربر استفاده می‌کند. این نوع رابط کاربری در محیط ترمینال اجرا می‌شود و به کاربر اجازه می‌دهد تا با استفاده از صفحه کلید، با برنامه تعامل داشته باشد.

برخلاف برنامه‌های خط فرمان ساده که فقط دستورات را می‌گیرند و خروجی می‌دهند، TUI می‌تواند ظاهری زیبا و تعاملی داشته باشد. شما می‌توانید منوها، لیست‌ها، و حتی جعبه‌های متنی را در محیط ترمینال نمایش دهید و با آن‌ها کار کنید.

```
Todos (4)
[ ] 1. Renew library books (Each 2 weeks)
>>[ ] 0- Workout (Daily)
[ ] 0. Brush teeth (Daily)
[x] 1. Dad's birthday (Reminder for 2024-05)

Todo dependencies
[ ] 0. Gym (Daily)
[ ] 0. Hiking (Daily)
[ ] 0. Boxing (Weekly)
```

شکل ۱: مثالی از یک برنامه مدیریت تسک در فضای TUI (لینک پروژه)

۲.۳ کتابخانه NCurses

برای پیاده‌سازی رابط کاربری متنی، از کتابخانه NCurses استفاده می‌کنیم. NCurses کتابخانه‌ای قدرتمند و چندپلتفرمی است که برای توسعه رابط‌های کاربری متنی در محیط‌های لینوکس و یونیکس مورد استفاده قرار می‌گیرد. این کتابخانه ابزارهای متنوعی برای مدیریت پنجره‌ها، نمایش منوها، کار با کیبورد و تعامل با ترمینال ارائه می‌دهد.

رفرنس‌های مفید برای شروع کار با کتابخانه NCurses:

- مستندات رسمی NCurses
 - مستندات در سایت لینوکس جورنال
 - سری مفید در یوتیوب
-

۳.۳ اهداف و نکات مهم پروژه

۱. **نوشتن کد تمیز و قابل فهم:**
کدهای نوشته‌شده باید خوانا، قابل درک و مرتب باشند. نوشتن کد تمیز امتیاز مثبت خواهد داشت.
 ۲. **ساختاردهی و فایل‌بندی پروژه:**
پیشنهاد می‌شود پروژه به بخش‌های مختلف تقسیم شود و هر بخش در فایل‌های جداگانه قرار گیرد تا قابلیت توسعه و دیباگ افزایش یابد.
 ۳. **درک کامل از نحوه کار با رشته‌ها و آرایه‌ها:**
برنامه شما باید بتواند تسک‌ها را به صورت موثری مدیریت کند و از داده‌های متنی و آرایه‌ها بهره‌برد.
-

۴ ساختار کلی پروژه

برنامه شما باید به طور کلی دارای ویژگی‌های زیر باشد (جلوتر توضیحات بیشتر در مورد هر یک از این ویژگی‌ها آمده است):

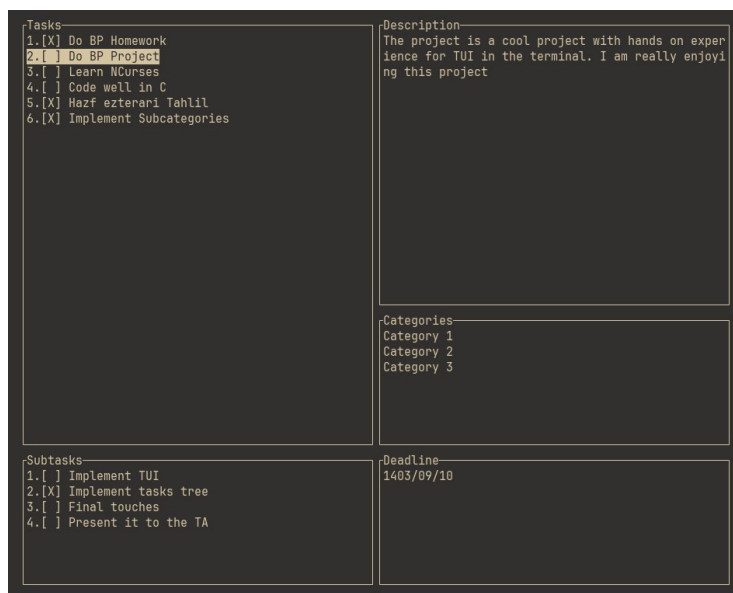
- داشتن یک رابطه کاربری TUI که بدون ایرادی قابل استفاده باشد.
- قابلیت دادن اولویت به هر تسک.
- قابلیت دسته‌بندی تسک‌ها.
- قابلیت افزودن یادداشت به هر تسک.
- قابلیت اضافه کردن و حذف کردن تسک‌ها.
- قابلیت تغییر حالت به «انجام شده» یا «انجام نشده» برای هر تسک.

- قابلیت اضافه کردن ددلاین به هر تسک.
- قابلیت مرتب سازی بر اساس اولویت، تاریخ ایجاد، وضعیت انجام و ...

۱.۴ ساختار TUI برنامه

شما بایستی سعی کنید که یک رابطه کاربری مانند عکس زیر پیاده سازی کنید.

- در این رابطه کاربری خانه بالا سمت چپ برای لیست کردن تسک ها است. در ابتدای هر تسک یک عدد برای اولویت نمایش داده شده که میتواند از ۱ الی ۹ باشد. سپس یک علامت «[]» یا «[X]» برای نشان دادن وضعیت انجام تسک قرار داده شده است. و بعد از آن نیز عنوان تسک نمایش داده شده است.
- در خانه بالا سمت راست نیز یک فضای متنی برای نمایش جزئیات تسک انتخاب شده است که همان توضیحات تسک است.
- در پایین خانه توضیحات تسک ها دسته بندی هایی که یک تسک دارد نمایش داده شده است.
- در پایین خانه دسته بندی ها یک لیست از تمامی دسته بندی ها نمایش داده شده است.
- در پایین خانه سمت چپ یک فضایی برای نشان دادن ساب تسک های یک تسک وجود دارد.



شکل ۲: یک نمونه ساختاری که میتوانید پیاده کنید

۲.۴ اولویت دهی به تسک ها

به هر تسک باید بتوانید یک عدد بین ۱ تا ۹ به عنوان اولویت اختصاص دهید. این عدد باید در ابتدای هر تسک نمایش داده شود و باید بتوانید تسک ها را بر اساس اولویت مرتب کنید. توضیحات بیشتر در مورد این ویژگی در بخش مرتب سازی تسک ها آمده است.

۳.۴ دسته بندی تسک ها

به هر تسک باید بتوانید هر چند تا که میخواهید دسته بندی اختصاص بدهید. این دسته بندی ها در خانه مشخص شده نشان داده میشوند.

۴.۴ افزودن یادداشت به تسک

هر تسک میتواند یک یادداشت داشته باشد. این یادداشت ها در خانه مشخص شده نمایش داده میشوند.

۵.۴ اضافه کردن و حذف کردن تسک ها

باید کاربر بتواند به راحتی هر چند تا که میخواهد تسک اضافه کند و یا تسک های موجود را حذف کند.

۶.۴ تغییر حالت به «انجام شده» یا «انجام نشده» برای هر تسک

کاربر بایستی بتواند برای هر تسک تغییر حالت انجام آن را به «انجام شده» یا «انجام نشده» دهد. وضعیت حالت انجام تسک در کنار عنوان تسک نمایش داده میشود. (مانند عکس داده شده)

۷.۴ اضافه کردن ددلاین به هر تسک

هر تسک میتواند یک ددلاین داشته باشد. این ددلاین ها در خانه مشخص شده نمایش داده میشوند. بایستی حتما برنامه بررسی کند که آیا ددلاین در حالت YYYY/MM/DD است یا خیر. و اگر نبود خطای مناسبی نمایش دهد.

۸.۴ مرتب سازی تسک ها

یکی از ویژگی های مهمی که برنامه شما باید داشته باشد این است که بتوانید تسک ها را بر اساس چندین معیار مختلف مرتب کنید. این معیار ها شامل:

- مرتب سازی بر اساس اولویت
- مرتب سازی بر اساس نزدیک ترین ددلاین
- مرتب سازی بر اساس حروف الفبا

۹.۴ سابتسک های ساده

هر تسک میتواند یک سری سابتسک داشته باشد. این سابتسک ها فقط یک عنوان دارند و فقط باید بتوانید آن ها را حذف کنید و اضافه کنید و وضعیت (انجام شده یا نشده) آن ها را تغییر دهید.

حالت پیشرفته تر این مورد در قسمت موارد امتیازی آمده است و اختیاری است.

۱۰.۴ قابلیت ذخیره و بازیابی تسک ها

شما بایستی با زدن یک کلید (پایین تعریف شده) بتوانید تمامی تسک ها را در یک فایل ذخیره کنید و با زدن یک کلید دیگر بتوانید تسک ها را از یک فایل بازیابی کنید. توجه کنید در هنگام باز شدن برنامه ابتدا باید تمامی تسک ها از فایل بازیابی شوند.

برای ذخیره سازی میتوانید از هر فرمتی استفاده کنید که JSON یک فرمت پیشنهادی است. برای استفاده از این فرمت میتوانید از کتابخانه های زیر استفاده کنید:

• Jansson

• cJSON

• json-c

۵ کلید های تعریف شده

برنامه شما بایستی یک سری کلید های مختص به خود را تعریف کند که کاربر بتواند با استفاده از آن ها به راحتی با برنامه تعامل داشته باشد. این کلید ها شامل:

• **کلید j:** این کلید برای حرکت به پایین در لیست تسک ها استفاده میشود. تسکی که الان انتخاب شده است باید با یک رنگ متفاوت نمایش داده شود.

• **کلید k:** این کلید برای حرکت به بالا در لیست تسک ها استفاده میشود. تسکی که الان انتخاب شده است باید با یک رنگ متفاوت نمایش داده شود.

• **کلید l:** در این حالت اشاره گر به روی سابتسک ها میرود و کاربر میتواند سابتسک ها را تغییر وضعیت (انجام شده یا نشده) کند و با کلید j و k بین سابتسک ها حرکت کند و کلید a و d برای اضافه کردن و حذف کردن سابتسک ها استفاده کند.

• **کلید h:** اگر کاربر اشاره گر روی سابتسک ها باشد با زدن این کلید به تسک اصلی برمیگردد.

• **کلید a:** با زدن این کلید کاربر میتواند یک تسک جدید اضافه کند.

• **کلید d:** با زدن این کلید کاربر میتواند تسک های انتخاب شده را حذف کند.

• **کلید e:** با زدن این کلید کاربر میتواند تسک های انتخاب شده را ویرایش کند.

• **کلید r:** با زدن این کلید کاربر میتواند توضیحات تسک انتخاب شده را ویرایش کند.

• **کلید n:** با زدن این کلید کاربر میتواند یک ددلاین جدید به تسک انتخاب شده اضافه کند.

- **کلید c:** با زدن این کلید، اشاره گر به دسته بندی ها میرود و کاربر میتواند با کلید a یک دسته بندی جدید برای تسک انتخاب شده اضافه کند و یا با کلید d یک دسته بندی را حذف کند و طبیعتا با کلید های j و k بین دسته بندی ها حرکت کند. با دوباره زدن این کلید اشاره گر به تسک اصلی ها برمیگردد.
- **کلید اسپیس:** با زدن این کلید کاربر میتواند وضعیت انجام تسک انتخاب شده را تغییر دهد. اگر تسک انجام شده بود به انجام نشده تغییر میکند و برعکس.
- **کلید q:** با زدن این کلید برنامه بسته میشود.
- **کلید w:** با زدن این کلید برنامه باید تمامی تسک ها را در یک فایل ذخیره کند.

۶ تحویل

پروژه را به صورت یک فایل فشرده به همراه فایل های مربوطه تحویل دهید. این فایل باید شامل:

- فایل های منبع برنامه
 - فایل های هدر
 - فایل های مورد نیاز برای کامپایل برنامه
- بعد از اتمام ددلاین پروژه، بایستی یک ارائه به یک تی ای از خود خواهید داشت و در آن کارکرد پروژه خود را به تی ای خود نشان خواهید داد و به سوالات مربوط به پروژه پاسخ خواهید داد. موفق باشید.

۷ موارد امتیازی

موارد امتیازی تنها در صورتی قابل قبول است که اصل پروژه به درستی پیاده سازی شده باشد و تقریبا نمره کامل از آن بگیرید.

- اضافه کردن قابلیت ساختار درختی سابتسک ها به تسک ها. یعنی هر تسک میتواند یک یا چند سابتسک داشته باشد یعنی تسک های کوچکتری که باید برای انجام تسک اصلی انجام شوند این سابتسک ها دقیقا از جنس تسک هستند یعنی این که برای آن ها هم میتونید اولویت، دسته بندی، یادداشت و ددلاین و ... اضافه کنید. در حقیقت تسک ها یک حالت درختی دارند که تسک اصلی ریشه و سابتسک ها بچه های آن هستند.

میتوان گفت که تسک ها همچین ساختار درختی دارند:

```
MainTask
├── SubTask1
│   ├── SubSubTask1
│   │   └── SubSubSubTask1
│   └── SubSubTask2
└── SubTask2
```



```
└─ SubTask3
└─ SubTask4
   └─ SubSubTasks3
```

• اضافه کردن آرگومان CLI لیست کردن تسک ها در برنامه. یعنی برنامه باید بتواند از خط فرمان لیست تسک ها را نمایش دهد.

– برای مثال اگر برنامه شما به نام todo است، اجرای دستور زیر باید لیست تسک ها را نمایش دهد:

```
./todo list
```

خروجی این دستور باید به صورت زیر باشد:

```
1. [ ] 1st task
2. [x] 2nd task
3. [ ] 3rd task
```

• قابلیت سرچ کردن تسک ها بر اساس عنوان یا دسته بندی و نشان دادن تسک های پیدا شده در خانه تسک ها.