پروژه 1

سیستم‌های عامل

Linux Shell

شما در این پروژه باید یک shell ساده یونیکس طراحی کنید که بتواند همانند یک shell معمولی دستورات وارد شده را اجرا و خروجی را به کاربر نمایش دهد.

توجه داشته باشید که پروژه شما علاوه بر دستورات معمول(مانند ls , cd, pwd و...) باید دستورات مشخص شده زیر که مربوط به پردازش متن میباشد را بتواند اجرا نماید.

- a : قسمت اول هر خط فایل متنی را که با اسپیس از بقیه خط جداشده را برمیگرداند(مثلا اگر در یک خط رشته good morning وجود داشته باشد باید good را برگرداند)
- b: رشته ای که تعداد بیشتری تکرار شده را بر میگرداند.

```
مثال

Hi
Hi
By
Hi
Morning
Result = Hi
```

- c: حذف کردن تمام empty space ها (space,/t,/n) از فایل متنی و سپس نمایش فایل
- d:نمایش خطوطی که کامنت نیستند(خطوط کامنت شده با # مشخص شده اند)
- f: نمایش تعداد خطوط فایل
- g: نمایش ده خط اول فایل

توجه داشته باشید که برای هر یک از دستورات بالا میتوانید به دلخواه یک اسم انتخاب کنید.

برای اجرا کردن دستورات نیاز است که با استفاده از fork یک فرایند مجزا اجرا نمایید و دستورات را در آن اجرا نمایید.

در ابتدای هر خط به عنوانpromptباید آدرس دایرکتوری فعلی باشد مانند مثال زیر(نحوه ی نمایش آن به عهده خودتان می باشد و فرمت خاصی ندارد). همچنین وقتی کاربر با استفاده از دستورcd به دایرکتوری دیگری رجوع میکند، بایدpromptآپدیت شود.

```
/home/amirmahdi/Desktop>
```

پیاده‌سازی قابلیت زیر به صورت نمره اضافه می‌باشد(میتوانید برای پیاده‌سازی آن درباره system signals تحقیق کنید) :

برنامه شما نباید با فشرده شدن ctrl+c بسته شود. در هنگام دریافت این ترکیب باید خط فعلی قطع وprompt جدید چاپ شده و آماده دریافت دستور جدید از کاربر باشد. (چاپ شدن یا نشدن ^c که کارکتر مربوط به ctrl +c اختیاری است).

```
/home/amirmahdi/Desktop> ls ^C
/home/amirmahdi/Desktop>
```

هنگامی که دستوری در حال اجراست صرفا اجرای آن دستور متوقف شود. برای خروج از برنامه باید دستور به شل داده شود.

در هنگام بروز خطا باید با استفاده از stderr آن خطا چاپ شود.

دستورات وارد شده باید در فایلی (آدرس دلخواه) به عنوان history ذخیره شوند و کاربر بتواند به آن‌ها دسترسی داشته باشد.

همچنین یک فایل README در کنار فایل خود داشته باشید و توضیحات کامل در نحوه کار shell و قابلیت‌ها و همچنین نامی که برای دستورات مشخص شده انتخاب کرده‌اید را بنویسید.

موارد نمره اضافه:

- با فشردن دکمه های بالا و پایین در history بچرخد تا دستوری که قبلا وارد کرده است را پیدا کند.
- پیاده‌سازی پایپلاین(هنگامی که چند دستور یا برنامه را پشت سرهم اجرا می کنیم و خروجی برنامه/دستور قبلی ورودی برنامه/دستور بعدی است یک pipeline ایجاد کرده ایم.)
- استفاده از گیت

توضیحات تکمیلی:

- زبان مورد استفاده میتواند C یا C++ باشد
- برای نوشتن این برنامه نیاز است درباره توابع زیر تحقیق نمایید و نحوه ی کارکرد آن ها را یاد بگیرید
  - ✗ fork
  - ✗ execvp
  - ✗ getpid
  - ✗ wait
  - ✗ pipe
- پروژه را میتوانید در قالب **گروه‌های دو نفره** پیاده‌سازی کنید.
- هرگونه شباهت بین دو گروه نمره 100- برای طرفین در پی دار.
- هنگام تحویل پروژه باید کامل به کد خود مسلط بوده و در صورت نیاز بتوانید تغییراتی در آن ایجاد کنید.
- **مهلت تحویل : جمعه 2 دی.23:55**

موفق باشید