



هدف از طرح این پروژه، شبیه سازی یک محیط ارتباط با کاربر تحت کنسول<sup>۱</sup> می باشد. محیط مد نظر چیزی شبیه ترمینال سیستم عامل لینوکس<sup>۲</sup> خواهد بود. برای پیاده سازی این نرم افزار کفایت یک دایرکتوری (فولدر) را به عنوان ریشه سیستم عامل فرضی خود استفاده کنید و کلیه دستورات در همین دایرکتوری پیاده خواهند شد. این شبیه ساز می بایست هر بار یک رشته از کاربر دریافت کرده و آن دستور را اجرا کند. این شبیه ساز از قسمت های متفاوتی تشکیل شده است که هر کدام به تفصیل توضیح داده خواهند شد.

## ✓ مدیریت کاربران (۴۰ درصد)

در این بخش شبیه ساز، مدیریت ورود و خروج کاربران، ذخیره سازی اطلاعات لازم و دیگر امکانات مورد نیاز پیاده سازی می شود. به ازای هر کاربر حداقل می بایست فیلد های نام کاربر، نام کاربری، رمز عبور و میزان قوت رمز عبور ذخیره شود.

- میزان قوت رمز عبور طبق قاعده ی زیر مشخص میشود. در سمت راست جدول وجود هر کاراکتر با شرط راست باعث اضافه شدن به اندازه درصد گفته شده به میزان قوت رمز عبور میشود. دقت شود که این عدد بیشتر از ۱۰۰ درصد نمیتواند باشد. برای مثال رمز عبور

دارای "Passw0rd" درصد قابل قبول  $3 + 2 * 6 + 4 = 19$  میباشد.

اجزای رمز عبور	میزان اضافه شدن به درصد قوت رمز عبور
حروف کوچک	۲ درصد
حروف بزرگ	۴ درصد
اعداد	۳ درصد
علائم نگارشی	۸ درصد
فاصله (اسپیس)	۱ درصد

همچنین برای هر کاربر بایستی مهلت کار با سیستم (به این معنا که این کاربر تا چه زمانی حق کار با سیستم را دارد. برای مثال اگر کاربری در زمانی بعد از تاریخ مشخص شده بخواهد به سیستم وارد شود سیستم از ورود او جلوگیری خواهد کرد)، سطح دسترسی کاربر (که می تواند ۰ یا ۱ باشد. صفر به معنای کاربر و یک به معنای مدیر سیستم می باشد) و تعداد دستورات اشتباه (در صورتی که دسترسی زده شود که جزو دستورات معرفی شده در زیر نباشد یک عدد اضافه میشود)، ذخیره شود.

با در نظر گرفتن ساختار بالا برای ذخیره سازی کاربران، در آغاز برنامه شبیه ساز می بایست منتظر ورود نام کاربری و رمز عبور از طرف کاربر باشد و در صورت صحت ورودی اجازه ورود بدهد. در غیر این صورت می بایست با نمایش پیام مناسب کاربر را مطلع سازد. پس از ورود شبیه ساز می بایست نام کاربری کسی که داخل سیستم است را همواره نمایش دهد. در صورتی که مهلت کار با سیستم برای کاربر پایان یافته، شبیه ساز نمی بایست اجازه ورود دهد. سپس کاربر دستورات زیر را میتواند دارد کند.

➤ دستور su : این دستور برای تغییر کاربر استفاده خواهد شد به صورتی که با ورود su username شبیه ساز میبایست پس از بررسی صحت رمز عبور به کاربر username تغییر کاربری دهد (مدیر برای ورود به دیگر حساب ها نیازی به وارد کردن رمز ندارد).

➤ دستور passwd : این دستور برای تغییر رمز عبور استفاده خواهد شد. به این صورت که هر کاربر با ورود این دستور می تواند رمز عبور خود را تغییر دهد.

لازم به ذکر است کاربر مدیر سیستم (سطح دسترسی ۱) می تواند رمز عبور تمامی کاربران را تغییر دهد به این صورت که می بایست دستور passwd را با نام

<sup>1</sup> Console

<sup>2</sup> <https://en.wikipedia.org/wiki/Linux>

کاربری کاربر مورد نظر وارد کند. دستور passwd با دریافت گزینه l- و دریافت زمان مهلت کار با سیستم کاربر را به زمانی دریافتی تغییر خواهد داد. (گزینه l- نیز فقط میتواند توسط مدیر سیستم اجرا شود):

### *passwd -l time username*

- روشی برای مدیریت سطوح دسترسی کاربران ارائه داده و پیاده‌سازی کنید به این صورت که با توجه به نظر سازنده (صاحب) فایل کاربران (گروه کاربران) خاصی اجازه دسترسی به فایل‌ها و دایرکتوری‌ها را داشته باشند.
- دستور create user: توسط مدیر انجام میشود و کاربری جدید را اضافه میکند و اطلاعات کاربر جدید را میگیرد و ذخیره میکند.
- ارتقاء سطح دسترسی: یک مدیر میتواند با کمک دستور *chgr username* کاربر با نام *username* را به مدیر تبدیل کند. منتها برای تبدیل یک کاربر باید شرایط زیر برقرار باشد:
  - کاربر حتما باید کاربر معمولی باشد
  - میزان قابل قبول بودن رمز عبور کاربر باید از ۷۵ درصد بیشتر باشد.
  - تعداد دستور اشتباهی که کاربر مورد نظر وارد کرده است از ۱۰ مورد نگذشته باشد.

**نکته:** میزان قوت رمز عبور بمنظور جلوگیری از نفوذ: هنگام تعیین رمز عبور برای کاربر (ساخت کاربر جدید یا تغییر پسورد) حتما باید از ۳۳ درصد بیشتر باشد!

### ✓ دستورات کار با فایل و دایرکتوری (۳۰ درصد)

- این مجموعه دستورات وظیفه ساخت و مدیریت فایل‌ها و دایرکتوری‌ها را برعهده خواهند داشت.
- در این شبیه ساز هر فایل مشخصه‌های نام فایل، سازنده فایل و آخرین زمان دسترسی به فایل است.
- دستورات این بخش عبارت اند از:
- دستور cd: این دستور محل جاری را به دایرکتوری خاصی تغییر می دهد (*cd dirname*) و در صورتی که به جای *dirname* از .. ( دو نقطه ) استفاده شود، محل جاری به یک دایرکتوری قبل خواهد رفت (آدرس ها هم میتوانند به صورت relative باشند هم absolute<sup>۳</sup>).
  - دستور pwd: این دستور نمایشگر محل جاری سیستم است و نمایش خواهد داد که شبیه ساز الان در کدام دایرکتوری است.
  - دستور mkdir: این دستور باید بتواند یک دایرکتوری جدید اضافه کند (*mkdir dirname*)
  - عملگر >: برای مثال در دستور (*Hello > filename*) عبارت Hello باید در فایل filename نوشته شود. حال اگر به جای Hello نام یک فایل قرار داشت کلیه محتویات آن فایل را داخل فایل filename می نویسد. (اول می بایست چک شود که Hello نام یک فایل در محل جاری است یا خیر، اگر خیر باید فایل ایجاد شود)
  - عملگر >>: این عملگر دقیقاً مانند > کار می کند با این تفاوت که این عملگر در ادامه ی فایل دوم خواهد نوشت.
  - دستور cat: با وارد کردن این دستور و نام فایل محتویات آن فایل نمایش داده خواهد شد (*cat filename*)
  - دستور rm: این دستور فایل را پاک می کند (*rm filename*). برای پاک کردن دایرکتوری ها باید از دستور *rm -r* استفاده شود.
- نکته:** برای دو دستور قبل، اگر فایل موجود نبود باید خطایی مناسب نمایش داده شود.
- دستور cp: این دستور یک فایل را در مکان دیگر کپی خواهد کرد. (*cp file1 file2*)
  - دستور mv: این دستور یک فایل را به مکان دیگر منتقل خواهد کرد. (*mv file1 file2*)

<sup>3</sup> <https://www.linux.com/blog/absolute-path-vs-relative-path-linuxunix>

\*نکته: در دو دستور قبل میتواند بجای اسم فایل ها آدرس آنان نوشته شود(`cp address1/file1 address2/file2`).

- دستور `ls`: محتویات دایرکتوری فعلی را نمایش میدهد(فایل ها و دایرکتوری ها)
- دستور `exif`: اطلاعاتی از قبیل سازنده فایل، زمان ساخت و آخرین تغییر ایجاد شده، مکان و ... را نمایش میدهد(`filenameexif`).

## ✓ قسمت ویرایشگر متن (۳۰ درصد)

هدف این قسمت، پیاده سازی یک ویرایشگر متن تحت خط فرمان است.

حداقل ها:

- ویرایشگر باید با دستور `myeditor` باز شده و محیط ویرایشگر را به کاربر نمایش دهد.
  - با زدن دستور `myeditor filename` باید محتویات فایلی به اسم `filename` را در صورت وجود داشتن آن فایل در محیط نمایشگر نمایش دهد. اگر فایل موجود نبود باید این نام فایل به عنوان نام پیش فرض برای ذخیره کردن در نظر گرفته شود. داخل محیط ویرایشگر:
  - کاربر باید بتواند دستور باز کردن یک فایل را به ویرایشگر بدهد. به این معنا که تمام محتویات آن فایل داخل ویرایشگر نمایش داده شود. برای مثال ویرایشگر با دریافت `CTRL+R` باید پس از دریافت نام فایل بتواند فایل را باز کرده و برای ویرایش آماده کند.
  - کاربر باید بتواند دستور ذخیره کردن را به ویرایشگر بدهد. یعنی متنی که داخل ویرایشگر موجود است را در یک فایل که نام آن را از کاربر میگیرد ذخیره کند.
  - کاربر باید بتواند از محیط ویرایشگر خارج شود. ویرایشگر قبل از خارج شدن از این کار مطمئن شود و اگر محتویات موجود در ویرایشگر هنوز در فایلی ذخیره نشده، قبل از خارج شدن، آن را ذخیره کند.
  - ویرایشگر باید امکان جستجوی یک رشته را داشته باشد.
  - ویرایشگر باید امکان کپی کردن یک قسمت از متن و `paste` کردن آن در مکانی دیگر از متن را داشته باشد.
  - پیام های مناسب باید در اجرای امکانات ویرایشگر به کاربر نمایش داده شود. کاربر باید توسط رابط کاربری طراحی شده توسط شما متوجه شود که باید برای رسیدن به هدف خود باید چه کار کند.
  - متن داخل ویرایشگر باید همیشه در پس زمینه قابل مشاهده باشد. به این معنا که پیام ها نباید تمام صفحه را بپوشاند.
  - اگر فایلی به عنوان فایل پیش فرض (فایل باز شده) داشته باشیم، نام پیش فرض برای ذخیره کردن باید نام این فایل باشد.
- ویرایشگر های `nano`<sup>۴</sup> و `vim`<sup>۵</sup> ویرایشگر های معروف تحت خط فرمان<sup>۶</sup> هستند. میتوانید از آنها الهام بگیرید و یا یکی از آنان را شبیه سازی کنید. میتوانید روش خودتان را برای اجرای امکانات ویرایشگر در نظر بگیرید ولی حداقل ها را باید رعایت کنید.

## ✓ نمره اضافی (نهایتاً ۲۰ درصد)

<sup>4</sup> [https://en.wikipedia.org/wiki/GNU\\_nano](https://en.wikipedia.org/wiki/GNU_nano)

<sup>5</sup> <http://www.vim.org/about.php>

<sup>6</sup> Command line

- دستور wc : این دستور گزارشی در مورد تعداد کلمات و تعداد خطوط یک فایل نمایش خواهد داد (*wc filename*)
- دستور diff: این دستور دو فایل را با هم مقایسه میکند و پیغامی نمایش میدهد(*diff file1 file2*).
- راه حلی برای ساخت فایل‌های پنهان (hidden) در شبیه ساز خود ارائه دهید. چگونه می‌توان فایل‌ها پنهان ساخت و با چه دستوری می‌بایست این فایل‌ها را نمایش داد و چگونه ؟
- دستوری برای یافتن یک فایل و یا دایرکتوری خاص در شبیه ساز خود طراحی و پیاده‌سازی کنید.
- یک راهنما برای دستورات خود بسازید به این صورت که با ورود یک دستور مانند help لیست تمامی دستورات شبیه ساز شما و توضیح مختصری نمایش داده شود و با ورود *help commandname* راهنمای مربوط به آن دستور نمایش داده شود.

### نکات مهم :

- تحویل اینترنتی این پروژه را میتوانید تا ساعت ۱۱:۵۹ یکشنبه ۶ بهمن ۱۳۹۸، در سامانه [lms.iut.ac.ir](http://lms.iut.ac.ir) انجام دهید. کدهای شما باید به صورت فایل زیپ و به صورت *stdnumber-firstname-lastname.zip* باشد.
- تحویل حضوری این پروژه در روزهای دوشنبه و سه شنبه ۷ و ۸ بهمن ۱۳۹۸ انجام خواهد شد.
- پیاده‌سازی کامل این پروژه ۱۰۰ نمره خواهد داشت و در صورت علاقه هر دانشجو می‌تواند ایده‌های خود را به کد اضافه کرده و متناسب با آن نمره ای اضافه بر نمره پروژه به آن‌ها تخصیص داده خواهد شد. (حداکثر این پروژه می‌تواند تا ۱۲۰ نمره داشته باشد).
- این پروژه به صورت انفرادی بوده و به تکالیف مشابه نمره ای تعلق نخواهد گرفت.
- در صورتی که سوالی داشتید میتوانید از طریق ایمیل [amirmnoohi@ec.iut.ac.ir](mailto:amirmnoohi@ec.iut.ac.ir) در ارتباط باشید.

**موفق باشید**