

به نام خدا



## مهارت‌های پیشرفته کار با کامپیوتر (بهار ۱۴۰۴)

### تمرین کامپیوتری ۱

مهلت ارسال: ۱۴۰۴/۰۱/۲۲

استاد درس: دکتر دوستی

دستیاران طراح: فرشته باقری - شهزاد ممیز

بازبینی: علی خرم فر

### قوانین و ملاحظات

#### نحوه ارسال تمرین:

- تمامی فایل‌ها باید در یک فایل فشرده با نام ECS-CA1-StudentID ارسال شوند.
- کدهای مربوط به هر بخش را با نام مناسب بر اساس جدول انتهایی همین فایل ذخیره کرده و همراه گزارش ارسال کنید.
- تمامی کدهای ارسال شده باید امکان اجرای مجدد داشته باشند. اگر تنظیمات خاصی برای اجرا نیاز است، آن را ذکر کنید.
- کدهای ارسال شده باید توسط خودتان اجرا شده باشند و نتایج اجرا در فایل ارسالی مشخص باشد.

#### رعایت اصول آکادمیک و صداقت علمی:

- این تمرین باید به صورت **فردی** انجام شود. هرگونه همکاری یا نوشتن تمرین به صورت گروهی ممنوع است.
- در صورت مشاهده تشابه در پاسخ، تمامی افراد درگیر نمره صفر دریافت خواهند کرد و موضوع به استاد گزارش خواهد شد.

#### استفاده از ابزارهای هوش مصنوعی:

استفاده از ابزارهایی مانند ChatGPT، Gemini، Copilot و موارد مشابه مجاز است، اما تحت شرایط زیر:

- نحوه استفاده از این ابزارها را در گزارش خود توضیح دهید (ابزارهای استفاده شده، کاربردهای مشخص و موارد مرتبط).
- تمامی پرامپت‌ها و لینک‌های استفاده شده را در انتهای گزارش قرار دهید.
- عدم ارائه این اطلاعات به منزله سرقت علمی محسوب شده و منجر به نمره صفر خواهد شد.

#### مهلت ارسال و جریمه تأخیر:

- امکان ارسال تمرین با **تأخیر تا ۲ روز** و به ازای **هر روز تأخیر ۱۰ درصد جریمه** وجود دارد.
- تأخیر به صورت ساعتی محاسبه شده و پس از دو روز تأخیر، تمرین پذیرفته نخواهد شد.

#### ارزیابی حضوری:

- ارزیابی تمرین به صورت حضوری انجام خواهد شد.
- محل ارزیابی: آزمایشگاه NLP، طبقه منفی یک، دانشکده مهندسی برق و کامپیوتر شماره ۲.

لطفاً پیش از شروع کار بر روی تمرین، به نکات زیر توجه فرمایید.

- حتماً ویدئوی راه اندازی کلاستر را به دقت مشاهده کنید و مطمئن شوید به کلاستر درس دسترسی دارید.
- آدرس های کلاسترهای درس به شرح ذیل است:

dml0: 172.18.32.200

dml1: 172.18.32.201

dml2: 172.18.32.202

dml3: 172.18.32.203

- برای راحتی در توسعه و تست کد، از ماشین مجازی لینوکس خود استفاده نمایید تا ترافیک کلاستر (به خصوص در ساعات آخر مهلت تمرین) افزایش نیابد. پس از اطمینان از عملکرد کد، می توانید آن را روی کلاستر اجرا کنید.
- در صورت بروز مشکل با ایمیل های زیر در ارتباط باشید:

سؤال ۱ و ۳: [fereshte12bagheri@gmail.com](mailto:fereshte12bagheri@gmail.com)

سؤال ۲ و ۴: [sh.momayez@gmail.com](mailto:sh.momayez@gmail.com)

### سؤال ۱. اجرای MOSH بدون دسترسی ROOT در سرور - ۱۰ نمره

فرض کنید mosh-server در سمت سرور نصب نشده است و دسترسی root نیز ندارید. چگونه می توانید از mosh برای برقراری استفاده کنید؟ مراحل کار را دقیق توضیح دهید.

### سؤال ۲. سیستم اتوماسیون و نظارت بر سرور راه دور مبتنی بر SSH با کنترل دسترسی مبتنی بر نقش (RBAC) - ۴۵ نمره (۱۰ نمره امتیازی)

در این قسمت، شما سیستمی طراحی خواهید کرد که امکان اجرای وظایف از راه دور، نظارت بر وضعیت سرور و انتقال فایل ها را از طریق SSH فراهم می کند. این سیستم شامل کنترل دسترسی مبتنی بر نقش (RBAC) خواهد بود تا نقش های مختلف مانند مدیر (Admin) و کاربر عادی (User) فقط به وظایف مجاز خود دسترسی داشته باشند.

#### راه اندازی سرور SSH

۱. نصب OpenSSH Server: ابتدا SSH Server را نصب کنید تا امکان اتصال از راه دور فراهم شود:

```
sudo apt update
```

```
sudo apt install openssh-server
```

مطمئن شوید که سرویس SSH در هنگام راه اندازی سیستم به طور خودکار اجرا می شود:

```
sudo systemctl enable ssh
```

```
sudo systemctl start ssh
```

۲. ایجاد حساب‌های کاربری و تخصیص نقش‌ها:

دو حساب کاربری ایجاد کنید:

- **مدیر: (admin\_user)** دسترسی کامل به سرور
- **کاربر عادی: (normal\_user)** دسترسی محدود برای نظارت

در محیط لینوکس می‌توانید از دستورات زیر استفاده کنید:

```
sudo adduser admin_user
```

```
sudo adduser normal_user
```

شما باید به کاربر مدیر اجازه دهید تا از دستوراتی مانند `sudo` برای انجام وظایف مدیریتی استفاده کند. این کار با اضافه کردن مدیر به گروه `sudo` انجام می‌شود.

## راه‌اندازی محیط Python و نصب وابستگی‌ها

۱. برای ارتباط با سرور از طریق SSH و انجام نظارت‌های سیستم، باید Python و چند کتابخانه را روی سیستم نصب کنید. مهم‌ترین کتابخانه‌ها عبارت‌اند از `paramiko` برای اتصال SSH و `psutil` برای نظارت بر وضعیت سیستم.

۲. احراز هویت مبتنی بر کلید SSH: به جای استفاده از رمز عبور برای ورود به سیستم، توصیه می‌شود از کلید SSH استفاده کنید. این روش امنیت بیشتری دارد. شما باید یک جفت کلید SSH ایجاد کنید و کلید عمومی را روی سرور قرار دهید تا به صورت ایمن به سرور متصل شوید.

به عنوان مثال می‌توانید از دستور زیر استفاده کنید:

```
ssh-keygen -t rsa -b 4096
```

## پیاده‌سازی کلاینت SSH با کنترل دسترسی مبتنی بر نقش

۱. تعریف نقش‌ها و مجوزها: شما باید سیستم را طوری طراحی کنید که نقش‌ها و مجوزها برای کاربران مختلف تعریف شود. به طور مثال، کاربر مدیر باید قادر به اجرای تمام دستورات سیستم باشد، در حالی که کاربر عادی تنها بتواند وضعیت `system health` را مشاهده کند و قادر به انجام عملیات حساس نباشد.

۲. تابع اتصال SSH به سرور: در این مرحله باید کدی بنویسید که به سرور متصل شود و از طریق SSH دستورات را ارسال کند. شما باید اطمینان حاصل کنید که اتصال SSH به صورت امن برقرار شده است و در صورت وجود خطا، پیام مناسبی برای کاربر نمایش داده شود.

۳. تابع اجرای دستورات از راه دور باتوجه به نقش: بسته به اینکه کاربر مدیر است یا کاربر عادی، باید دستورات مختلفی را اجرا کنید. برای مدیر، دستورات سیستم مانند پشتیبان گیری و به روزرسانی باید مجاز باشد، اما برای کاربر عادی، تنها دستورات محدود به نظارت بر وضعیت سیستم مجاز است.

۴. نظارت بر health سرور: شما باید اسکریپتی بنویسید که بتواند وضعیت سیستم مانند استفاده از CPU، حافظه و فضای دیسک را بررسی کرده و آن را نمایش دهد. این اسکریپت می تواند از کتابخانه psutil برای دریافت اطلاعات سیستم استفاده کند.

### پیاده سازی اتوماسیون سرور (پشتیبان گیری و به روزرسانی سیستم)

شما باید یک اسکریپت بنویسید که به صورت دوره ای از دایرکتوری های مهم مانند /etc و /home پشتیبان گیری کند. این اسکریپت باید فایل های پشتیبان را فشرده کرده و در یک دایرکتوری مشخص ذخیره کند. اسکریپت شما باید run\_interval و output\_dir را به عنوان کانفیگ قبول کند و بتوان این تنظیمات را بدون تغییر کد اعمال کرد و دوباره سیستم را راه اندازی کرد.

### انتقال فایل ایمن با SFTP

۱. آپلود و دانلود فایل ها: کاربران مدیر باید قادر به بارگذاری و دانلود فایل ها باشند. در اینجا کد مربوطه را بنویسید به نحوی که کاربران مدیر بتوانند فایل ها را upload و download کنند. کد شما می تواند با دستورات زیر اجرا شود: (در هر نوع پیاده سازی حتماً باید دو آدرس به عنوان آرگومان داده شود).

<Download/Upload>\_file client local\_path remote\_path

### امتیازی:

- به انتخاب خود یکی می توانید یکی از دو بخش امتیازی زیر را پیاده سازی کنید. انجام هر دو بخش نمره اضافی نخواهد داشت.
- به منظور پیگیری اقدامات کاربران (دستورات اجرا شده، فایل های آپلود یا دانلود شده) سیستم گزارش گیری (logging) را در سرور و کلاینت پیاده سازی کنید.
- فیچری را پیاده سازی کنید که به کاربر عادی اجازه می دهد در یک دایرکتوری مشخص فایل ها را آپلود و از آن دانلود کند. در صورتی که کاربر عادی path download/upload غیرمجاز وارد کند، ارور مسیج مناسب نشان دهید.

### سؤال ۳. مانیتورینگ سلامت سیستم از زمان بوت و مدیریت پردازش‌ها - ۴۵ نمره (۱۰ نمره امتیازی)

۱. شما باید سیستمی طراحی کنید که از لحظه‌ی بوت شدن سیستم، منابع سخت‌افزاری مانند CPU، RAM و فضای دیسک را پایش کند. این سیستم باید بتواند در صورت عبور از آستانه‌های تعیین‌شده، هشدار ارسال کرده و پردازش‌های پرمصرف را به حالت تعلیق (Suspend) درآورد تا عملکرد سیستم پایدار بماند. ر صورت عبور میزان مصرف منابع از حد تعیین‌شده، سیستم باید دو اقدام انجام دهد:

۲. ارسال هشدار (Alert): این هشدار می‌تواند به‌صورت پیام در لاگ سیستم، ایمیل، یا نوتیفیکیشن CLI باشد.
۳. مدیریت پردازش‌های پرمصرف: پردازش‌هایی که باعث افزایش مصرف منابع شده‌اند، باید به حالت تعلیق (Suspend) درآیند تا عملکرد کلی سیستم تحت‌تأثیر قرار نگیرد.

#### ۱. پیاده‌سازی مانیتورینگ سلامت سیستم:

- استفاده از اسکریپت بررسی سلامت سیستم (System Health) که در سوال قبل طراحی شده است.
- پایش دوره‌ای میزان مصرف CPU، RAM و فضای دیسک با استفاده از کتابخانه psutil در Python.
- تعریف آستانه‌های پیش‌فرض (مثلاً ۸۰٪ برای CPU و RAM) و امکان دریافت مقادیر جدید از طریق آرگومان‌های خط فرمان.

#### ۲. اجرای برنامه از زمان بوت:

- ثبت اسکریپت مانیتورینگ به‌عنوان یک سرویس systemd یا استفاده از rc.local برای اجرای خودکار در هنگام بوت.
- اطمینان از اینکه سرویس در صورت قطع‌شدن، مجدداً راه‌اندازی می‌شود.

#### ۳. ارسال هشدار در صورت عبور از آستانه‌ها:

- ثبت هشدارها در یک فایل لاگ در `/var/log/system_monitor.log`.
- نمایش پیام هشدار در خط فرمان.

**نمایش هشدار در خط فرمان:**

#### ۴. تعلیق پردازش‌های پرمصرف:

- شناسایی پردازش‌هایی که بیشترین مصرف CPU یا RAM را دارند.
- استفاده از دستور `kill -STOP <PID>` برای متوقف کردن موقت پردازش‌های پرمصرف.
- نمایش لیستی از پردازش‌های متوقف‌شده و امکان Resume آن‌ها با `kill -CONT <PID>`.

#### امتیازی:

- امکان ذخیره گزارش‌های مصرف منابع و پردازش‌های متوقف‌شده در یک فایل CSV برای تحلیل‌های بعدی.
- نمایش وضعیت سیستم به‌صورت گرافیکی با استفاده از Matplotlib یا Dash.

## سؤال ۴. بوت کردن سیستم و استفاده از GRUB – ۲۰ نمره

در این قسمت، شما با نقش و عملکرد بوت لودرها آشنا خواهید شد و یاد خواهید گرفت که چگونه GRUB را پیکربندی کنید، تنظیمات آن را تغییر دهید و تأثیر این تغییرات را بر فرایند بوت سیستم مشاهده و تحلیل کنید. همچنین، درک دقیقی از نحوه بارگذاری کرنل و تفاوت‌های بین BIOS و UEFI به دست خواهید آورد.

### سؤالات تشریحی:

پاسخ سؤالات تشریحی را به صورت کامل در فایل گزارش پروژه خود بنویسید.

مکانیزم‌های بوت لودر GRUB را بنویسید.

راجع به BIOS و UEFI توضیح دهید.

راجع به چگونگی بارگذاری و اجرای کرنل‌ها در سیستم‌های عامل بنویسید.

با توجه با تصویر زیر راجع به مفاهیم زیر توضیح دهید:

- Partition table
- Boot signature
- Winhex program



### پیکربندی GRUB و مشاهده نتایج

- زمان انتظار بوت را تغییر دهید و مشاهده کنید که چگونه روی روند بوت تأثیر می‌گذارد.
- سیستم‌عامل جدیدی را به منوی بوت اضافه کنید (اگر از ماشین مجازی استفاده می‌کنید، یک توزیع لینوکس سبک مانند Alpine را در کنار سیستم‌عامل اصلی نصب کنید).
- یک ورودی سفارشی در GRUB ایجاد کنید که یک اسکریپت را هنگام بوت اجرا کند (مثلاً اسکریپتی که اطلاعات سیستم را نمایش دهد).

## سفارشی سازی GRUB

ظاهر منوی GRUB را تغییر دهید (تغییر تصویر پس زمینه، رنگ بندی و فونت ها).

### تحلیل عملکرد سیستم پس از تغییرات

- با استفاده از ابزار `systemd-analyze`، زمان بوت سیستم را قبل و بعد از تغییرات اندازه گیری کنید و نتایج را مقایسه کنید.
- از دستورات زیر استفاده کنید و خروجی آن ها را در گزارش خود ثبت کنید:

`systemd-analyze`

`systemd-analyze blame`

برای ارزیابی این قسمت، فایل های زیر را ارائه دهید:

- پاسخ سؤالات تشریحی
- تغییراتی که در GRUB اعمال کردید (همراه با توضیح و دستورات استفاده شده)
- تحلیل عملکرد بوت سیستم قبل و بعد از تغییرات
- اسکرین شات های مربوط به اجرای دستورات و منوی سفارشی شده GRUB
- در صورت امکان، یک ویدئوی کوتاه از اجرای تغییرات و تأثیر آن ها بر فرایند بوت (اگر امکان رکورد ویدئو را نداشتید، در گزارش کار خود با استفاده از اسکرین شات تمام مراحل انجام شده را نشان دهید).

## نحوه تحویل تمرین کامپیوتری ۱

فایل‌ها را به صورت زیر نام‌گذاری کرده و همه را در یک فایل zip در سامانه ارسال کنید.

نام فایل‌ها	بخش	سؤال
<p>تمامی موارد مربوط به این قسمت را در</p> <p>یک پوشه با نام P1 قرار دهید:</p> <p>پاسخ سؤال را در یک فایل PDF</p> <p>بنویسید.</p>	-	۱
<p>تمامی موارد مربوط به این قسمت را در</p> <p>یک پوشه با نام P2 قرار دهید:</p> <p>تمام کدها</p> <p>نحوه اجرای هر یک از بخش‌های</p> <p>خواسته شده در صورت سؤال را در یک</p> <p>فایل PDF گزارش کنید.</p>	تمام بخش‌ها	۲
<p>تمامی موارد مربوط به این قسمت را در</p> <p>یک پوشه با نام P3 قرار دهید:</p> <p>تمام کدها</p>	تمام بخش‌ها	۳



نحوه اجرای هر یک از بخش‌های خواسته شده در صورت سؤال را در یک فایل PDF گزارش کنید.		
تمامی موارد خواسته شده در صورت سؤال را در یک پوشه با نام P4 قرار دهید.	تمام بخش‌ها	۴

دقت کنید که تمام دستورات مورد نیاز برای اجرای برنامه‌های خواسته شده باید در فایل PDF آورده شوند و این دستورات باید در پوشه top level مربوط به هر سؤال پروژه قابل اجرا باشند.