

## نحوه ارزیابی پایانی درس آزمایشگاه سیستم عامل

لازم است این نکته را متذکر شوم که به این ارزیابی به دید نمره یا درس نگاه نکنید (البته که برای درس هست و نمره دارد)، اما به این دید نگاه کنید که قرار است شما را با حوزه‌ای از کامپیوتر آشنا نماید که ان‌شالله در آینده برای شما مفید خواهد بود.

ارزیابی پایانی شامل ۳.۵ نمره است که به دو بخش ارائه انفرادی و پروژه گروهی تقسیم می‌شود.

### ارائه فردی (۱ نمره):

درس آزمایشگاه سیستم عامل یک درس عملی است، لذا در ارائه‌های فردی نیز بایستی این موضوع مد نظر قرار گیرد. این ارائه شامل یک بخش توضیحی و یک بخش عملی است که در ۲۵ الی ۳۰ دقیقه برگزار خواهد شد. در بخش توضیحی مفاهیم مرتبط با موضوع انتخابی را مطرح نموده و سایر دانشجویان را با موضوع مورد نظر آشنا نمایید. سعی نمایید در صورت امکان از کاربردهای موضوع مورد نظر در دنیای واقعی هم مثال‌هایی را ذکر نمایید (اختیاری). سپس در بخش عملی، بسته به موضوع انتخابی، بخش‌های مختلف کد یا اسکریپت از قبل آماده‌ای را توضیح داده و آن را اجرا می‌نمایید. توجه شود که در بخش عملی ارائه‌ها تمرکز بر روی استفاده از ترمینال است و تا حد امکان بایستی از استفاده از ابزارهای گرافیکی خودداری شود.

حوزه	موضوع	توضیحات
Administrative	1-Shell Scripting	
	2-SSH + Network Services	یکی از سرویس‌های شبکه‌ای مانند راه‌اندازی اولیه یک سرور HTTP، FTP، Mail یا غیره
	3-File System Management	mount کردن، استفاده از fdisk و mkfs و غیره
Application Programming	این حوزه مورد توجه این درس نیست	
System Programming	1-Daemon	
	2-Raw Socket	
	3-Virtual Network Interface (tun/tap)	
	4-IPC (except pipe)	
	5-Static and Dynamic Library	
	6-Selected topics from reference book	می‌توانید برخی از مباحث کتاب مرجع را که در ادامه مباحث مطرح شده در کلاس است، انتخاب نمایید (هماهنگ کنید)
Kernel Module Development	1-Kernel Module Development with emphasis on user/kernel space interfaces	
	2-Device Driver Development	
Kernel Development	1-Kernel Internals	درباره یکی از زیرسیستم‌های هسته مانند بخش شبکه، مدیریت حافظه، مدیریت پردازش یا غیره
Tools	1-GDB for debugging	
	2-makefile + overview of Cmake and Autotools	
	3-Program Format in Linux (ELF)	

دسته‌بندی موضوعات همانند دسته‌بندی ارائه‌شده در جلسه اول کلاس است تا نه تنها جایگاه هر زمینه در لینوکس بیشتر مشخص شود، بلکه در صورتی که علاقمند به یادگیری مباحث بیشتری بودید، بتوانید در زمینه مورد نظر تمرکز نمایید.

## پروژه گروهی (۲۰۵نمره):

توجه: پیاده‌سازی‌ها حتماً بایستی در زبان C باشد و از کتابخانه‌های آماده (به جز کتابخانه C) استفاده نشود.

توجه: دوباره تأکید می‌کنم که پیاده‌سازی‌ها حتماً بایستی در زبان C باشد و از کتابخانه‌های آماده (به جز کتابخانه C) استفاده نشود.

پیاده‌سازی یک وب‌سرور ساده در فایل ضمیمه آورده شده است. در صورتی که با موفقیت آن را ساخته و اجرا نمایید، صفحه زیر را در مرورگر خود

مشاهده خواهید نمود:



در این پروژه قصد داریم تا کد وب‌سرور ساده ارائه‌شده را به نحوی تغییر دهیم که درخواست کاربران در پدازه‌ها و نخ‌های جداگانه رسیدگی شود. پیاده‌سازی وب‌سرور ساده بایستی به نحوی باشد که در **زمان کامپایل** بتوانیم برنامه را به صورت چندپدازه‌ای یا چندنخی بسازیم، به عبارت دیگر نبایستی دو پروژه مجزا ایجاد نمایید، بلکه تمامی کدها در یک پروژه بایستی قرار داشته باشد و در زمان کامپایل با توجه به نحوه ساخت، **فقط** تکه کدهای مورد نیاز کامپایل شده و سایر بخش‌ها کامپایل نشوند. بخش‌هایی که میان هر دو پیاده‌سازی مشترک هستند، بایستی در کد **یک مرتبه** آورده شوند (راهنمایی: درباره Conditional Compilation در زبان C تحقیق کنید).

### قابلیت‌های مد نظر برای پروژه عبارتند از:

- ۱- ایجاد تغییرات لازم برای چندپدازه‌ای و چندنخی کردن سرور با اعمال محدودیت بر روی تعداد پدازه‌ها و نخ‌هایی که همزمان فعالیت می‌نمایند، به عنوان مثال می‌توانید از یک استخر نخ برای محدود کردن تعداد نخ‌ها استفاده نمایید.
- ۲- خواندن تنظیمات سرور همچون شماره پورت، حداکثر تعداد پدازه‌ها یا نخ‌های همزمان فعال، مسیر قرارگیری محتوی‌ها و ذخیره‌سازی گزارش‌ها از یک فایل تنظیمات که در مسیر از قبل مشخصی قرار گرفته است (مثلاً `/etc/simpleWebServer/server.conf`).
- ۳- ثبت گزارش رسیدگی به درخواست تمامی کاربران همچون زمان، محتوی مورد درخواست و مرورگر کاربر در پدازه پدر (ارتباط میان پدازه‌ای) یا نخ اصلی. دقت نمایید که تمامی کارهای مرتبط با رسیدگی به درخواست در پدازه یا نخ جداگانه انجام می‌شود و اطلاعات استخراج‌شده از درخواست در اختیار پدازه پدر یا نخ اصلی قرار می‌گیرد. در انتها، زمانی که سرور با فشردن `ctrl+c` خاتمه می‌یابد، این گزارش‌ها توسط پدازه پدر یا نخ اصلی چاپ می‌شود. گزارش‌های بدست‌آمده را برای تحلیل‌های بعدی نیز در یک فایل جداگانه که مسیر آن در فایل تنظیمات مشخص شده است، به صورت تجمعی (یعنی با اجرای مجدد سرور و ذخیره گزارش‌های جدید، گزارش‌های قبلی از بین نروند) نگهداری نمایید.

۴- توانایی ارسال صفحات استاتیک HTML مختلف. با توجه به مسیر موجود در فایل تنظیمات، محتوای درخواستی بازگردانده می‌شود. در صورتی که صفحه درخواستی موجود نباشد، یک صفحه پیش فرض خطا بازگردانده شود و در صورت عدم وجود فایل‌های جانبی همچون تصویر، چیزی بازگردانده نمی‌شود. لازم نیست فایل‌های HTML پیچیده باشند و می‌توانید از فایل HTML ساده استفاده نمایید. دقت نمایید که حداقل دو صفحه HTML که حاوی حداقل یک تصویر در هر یک از آن‌ها باشد، ایجاد نمایید. در صفحه اول لینکی قرار دارد که با کلیک بر روی آن صفحه دوم دیده خواهد شد.

۵- نوشتن یک makefile برای ساخت پروژه که شامل دستورات زیر است:

عملکرد	دستور
ساخت پروژه به صورت چندپردازه‌ای	make TYPE=multiProcess
ساخت پروژه به صورت چندنخی	make TYPE=multiThread
نصب برنامه در مسیر پیش فرض برنامه‌ها در لینوکس تا برنامه در ترمینال شناخته شود + ایجاد یک فایل تنظیمات اولیه در مسیر <code>/etc/simpleWebServer</code>	make install
پاکسازی پروژه از فایل‌های ساخته شده	make clean

۶- تهیه گزارشی درباره سیستم، نحوه ساخت و استفاده از آن. برای تهیه گزارش لازم نیست از قالب‌های رسمی گزارش استفاده نمایید، بلکه می‌توانید یک فایل README به پروژه اضافه نموده و اطلاعات را در آن وارد نمایید.

۷- (اختیاری، یک نمره اضافه) وب سرور خود را برای هر دو سیستم عامل لینوکس و ویندوز پیاده‌سازی نمایید. دقت نمایید که در این بخش نیز هدف ایجاد دو پروژه مجزا نیست، بلکه همانند قابلیت چندپردازه‌ای و چندنخی، تمامی کدها در یک پروژه قرار داشته و در زمان کامپایل می‌توانیم آن را برای سیستم عامل مقصد بسازیم. در این پیاده‌سازی نیز بایستی تا حد امکان، بخش‌های مشترک فقط یک مرتبه در کد آورده شوند. برای ساخت وب سرور که مثلاً به صورت چندنخی در سیستم عامل ویندوز کار کند، می‌توانیم از دستور زیر استفاده نماییم:

make PLATFORM=windows TYPE=multiThread

## تذکرات:

۰- مواردی اولیه همچون ساختار مناسب پروژه (قرارگیری فایل‌های سرآیند، کد و ساخته شده در پوشه‌های جداگانه)، سبک کدنویسی و بررسی خروجی توابع رعایت شود.

۱- ارائه پروژه‌ها به صورت حضوری (اگر شرایط فراهم نباشد، مجازی) است و همه اعضای گروه بایستی به **همه** بخش‌های پیاده‌سازی مسلط باشند، پس کارهای خود را به سایر اعضا گروه خودتان توضیح دهید.

۲- می‌توانید از تمامی مطالبی که در همه منابع از جمله اینترنت پیدا می‌کنید، استفاده نمایید.

۳- تقلب نکنید، نه چون من خیلی حرفه‌ای تقلب می‌کنم که دیگران نمی‌فهمند، بلکه چون خدا و خودمان می‌دانیم که تقلب کردیم و این کار، کلاه گذاشتن سر خودمان هست، حتی اگر نمره کاملی از پروژه بگیریم.

۴- اگر سؤالی درباره پروژه داشتید، می‌توانید در گروه یا ایمیلی ([memarnejadm74@gmail.com](mailto:memarnejadm74@gmail.com)) بپرسید.

موفق باشید

مهران معمارنژاد