

## امتحان درس آزمایشگاه سیستم عامل

قبل از شروع امتحان با دقت شرط‌های زیر را مطالعه نمایید:

- نمره‌ی امتحان از صد محاسبه می‌گردد و با توجه به درستی جواب، زمان ارائه‌ی آن و تعداد سؤال‌های پرسیده شده در زمان امتحان محاسبه خواهد شد.
- زمان امتحان هشتاد دقیقه است.
- تا دقیقه‌ی پنجم، فقط به سؤال‌هایی که در مورد صورت مسئله‌های امتحان هستند، پاسخ داده می‌شود. این سؤال‌ها در محاسبه‌ی نمره لحاظ نمی‌شوند.
- پس از دقیقه‌ی چهارم، هر ده دقیقه چهار نمره از افرادی که جواب را تمویل نداده باشند، کم می‌شود.
- تا دقیقه‌ی چهارم به هیچ سؤال‌ی پاسخ داده نمی‌شود. از دقیقه‌ی چهارم، هر دانشجو می‌تواند حداکثر سه سؤال بپرسد. با پرسیدن هر سؤال، پنج نمره از نمره‌ی امتحان آن دانشجو کاسته می‌شود.
- سؤال‌های پرسیده شده می‌توانند در مورد رفع خطاهای مترجم، رفع خطاهای دستورهای پوسته، یا توضیح کلی در مورد رویکرد مل باشند. به سؤال‌هایی که مستقیماً در مورد جواب باشند، پاسخ داده نمی‌شود. برای مثال، سؤال «چه تغییری باید در این فایل بدهم؟»، یک سؤال مجاز نیست.
- استفاده از منابعی چون دستور و گزارش آزمایشگاه‌ها آزاد است.
- هر گونه ارتباط با افراد داخل آزمایشگاه و خارج از آن مجاز نمی‌باشد.
- در هر مسئله، جواب باید در فایل‌ی با نام شماره‌ی دانشجویی و با پسوند «.sh» یا «.c» تمویل داده شود.

## گروه اول

اسکرپت پوسته‌ای بنویسید که همه‌ی فایل‌های با پسوند `.txt` را در شافه‌ی جاری و زیر شافه‌های آن بررسی کند و نام آنهایی را چاپ کند که تعداد فتهای آن برابر پارامتر ورودی باشد.

در فایل `oslabe1.c` سه بند بسازید که `thread1()`، `thread2()` و `thread3()` را به صورت موازی فراخوانی کنند. سپس این فایل را به صورتی تغییر دهید که پس از اینکه اولین بند مقدار متغیرهای `first` و `first_id` را تغییر داد، بند دیگری این دو متغیر را تغییر ندهد.

## گروه دوم

ر.م. قرار است برای مدتی کوتاهی کامپیوتر شخصی‌اش را در اختیار یکی از دوستانش قرار دهد. اما نمی‌خواهد دوستش به اطلاعاتش (از جمله کد سیستم عامل غوفو) دسترسی داشته باشد. از این رو قصد دارد با دستور `base64` برفی از فایل‌هایش را کدگذاری کند. این دستور متنی را که از ورودی استاندارد دریافت می‌کند به شکل `base64` کدگذاری می‌نماید و در خروجی استاندارد چاپ می‌کند. برای کمک به ر.م. اسکرپتی بنویسید که محتویات همه‌ی فایل‌های شافه‌ی جاری و زیرشافه‌های آن را که پسوند `.c` دارند با این دستور کدگذاری نماید.

در فایل `oslabe2.c` چهار تابع فراخوانی می‌شوند. برای بهره‌برداری از هسته‌های پردازنده‌اش، ر.م. می‌خواهد هر یک از این توابع در یک پردازهی مجزا اجرا شود ولی در هر لحظه نباید بیشتر از دو پردازه اجرا شوند. به او کمک کنید.

## گروه سوم

ر.م. می‌خواهد بداند کدام یک از فایل‌هایش توسط پردازهای باز شده است. با نوشتن اسکرپت پوسته‌ای به او کمک کنید. اسکرپت باید آدرس یک شافه را بگیرد و همه‌ی فایل‌های موجود در آن شافه و زیر شافه‌های آن را که توسط حداقل یک پردازه باز شده‌اند چاپ کند. می‌توانید از دستور `fuser` کمک بگیرید.

در فایل `oslabe3.c` سه بند ساخته می‌شوند. ر.م. می‌خواهد این فایل به صورتی تغییر دهد که فراخوانی تابع `B` توسط بند سوم، پس از پنج بار فراخوانی `A` توسط دو بند اول و دوم انجام شود. به او کمک کنید.

## گروه چهارم

ر.م. مشکلات گزارش شده توسط کاربران سیستم عامل غوفو را در تعدادی شافه نگه می‌دارد (اطلاعات هر مشکل در یک شافه‌ی مجزا قرار دارد). در شافه‌ی مربوط به مشکلات مهم، فایل‌ی با نام urgent وجود دارد (ممتویات این فایل مهم نیست). ر.م. به فهرست مشکلات مهم امتیاج دارد. اسکریپت پوسته‌ای بنویسید که با گرفتن یک شافه، همه‌ی زیر شافه‌های آن را که حاوی فایل urgent هستند نمایش دهد.

در فایل `oslabe2.c` چهار تابع فراخوانی می‌شوند. ر.م. می‌خواهد هر یک از این توابع در یک پردازش مجزا اجرا شود ولی پردازش‌های تابع B باید بلافاصله پس از پایان پردازش‌های تابع A و پردازش‌های تابع D باید بلافاصله پس از پایان پردازش‌های تابع C شروع شوند. همه‌ی پردازش‌ها باید توسط بند اصلی (که اجرای تابع `main()` را آغاز می‌کند) ایجاد شوند. یادآوری می‌شود که تابع `wait()` شناسه‌ی فرزندی که فاتمه یافته است را بر می‌گرداند.

## گروه پنجم

ر.م. قصد دارد فهرست همه‌ی شافه‌هایی که شامل زیر شافه‌های با نام `git` هستند را بدست بیاورد. اسکریپتی بنویسید که زیرشافه‌های شافه‌ی جاری و زیرشافه‌های آنها را بررسی کند و آدرس آنهایی را چاپ کند که شامل زیر شافه‌های با نام `git` هستند.

در فایل `oslabe5.c` سه تابع فراخوانی می‌شوند. ر.م. می‌خواهد زمان اجرای سه تابع را مقایسه کند. این فایل را به صورتی تغییر دهید که هر یک از این توابع در یک پردازش جدید به صورت همروند اجرا شود. سپس، نام پردازش‌های را چاپ کند که زودتر از همه فاتمه می‌یابد. یادآوری می‌شود که تابع `wait()` شناسه‌ی فرزندی که فاتمه یافته است را بر می‌گرداند.

## گروه ششم

ر.م. می‌خواهد فهرست همه‌ی پردازش‌های `Zombie` سیستم عامل را بدست آورد. اسکریپتی بنویسید که شناسه‌ی همه‌ی پردازش‌های `Zombie` را به کمک شافه‌ی `/proc` نمایش دهد. برای یک پردازش با شناسه‌ی 123 یک شافه در آدرس `/proc/123` ساخته می‌شود. اگر فایل `status` موجود در این شافه شامل رشته‌ی `zombie` باشد، پردازش‌های 123 یک `Zombie` است.

در فایل `oslabe5.c` سه تابع فراخوانی می‌شوند. ر.م. می‌خواهد زمان اجرای سه تابع را مقایسه کند. این فایل را به صورتی تغییر دهید که هر یک از این توابع در یک بند جدید به صورت همروند اجرا شود. سپس، نام پردازش‌های را چاپ کند که زودتر از همه فاتمه می‌یابد.

## گروه هفتم

ر.ه. نیاز دارد شافه‌هایی را بیابد که مداخل یک فایل با پسوند h دارند ولی فایلی با پسوند c ندارند. اسکریپت پوسته‌ای بنویسید که در زیر شافه‌های شافه‌ی جاری جستجو کند و آدرس همه‌ی شافه‌هایی که این ویژگی را دارند چاپ کند.

ر.ه. می‌فواهم با استفاده از لوله دو پردازنده را همگام کند. در فایل `oslabe7.c` پنج بار تابع A و پنج بار تابع B فراخوانی می‌شوند. این فایل را به شکلی تغییر دهید که فراخوانی‌های تابع B در یک پردازنده‌ی جدید به صورت موازی با پردازنده‌ی اصلی انجام شوند. سپس، با استفاده از لوله کاری کنید که فراخوانی i-A ب بلافاصله بعد از پایان فراخوانی i-A آغاز شود.