

## امتحان درس آزمایشگاه سیستم عامل

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

- نمره‌ی امتحان از صد مناسبه می‌گردد و با توجه به درستی جواب، زمان ارائه‌ی آن و تعداد سؤال‌های پرسیده شده در زمان امتحان مناسبه 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 مناسبه‌ی 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 شماره‌ی دانشجویی و با پسوند «.sh» یا «.c» تمویل داده شود.

## گروه اول

اسکرپت پشته‌ای بنویسید که همه‌ی فایل‌های شافه‌ی جاری و زیر شافه‌های آن را که پسوند آنها «.exe» است، پاک کند.

در فایل `oslabe1.c` یک ریسمان سافته می‌شود. سپس، ریسمان اصلی و ریسمان جدید به صورت موازی اجرا می‌شوند. از دسترسی‌های همزمان به متغیرها در این دو ریسمان ممانعت کنید.

## گروه دوم

اسکرپت پشته‌ای بنویسید که فایل‌های فالی شافه‌ی جاری و زیر شافه‌های آن را پاک کند.

در فایل `oslabe2.c` یک ریسمان سافته می‌شود. این فایل را به شکلی تغییر دهید که تابع `compute2()` در ریسمان جدید اجرا شود (همزمان با `compute1()` در ریسمان اصلی) و فراخوانی تابع `compute3()` در ریسمان اصلی پس از اتمام فراخوانی تابع‌های `compute1()` و `compute2()` شروع گردد.

## گروه سوم

اسکرپت پشته‌ای بنویسید که تعداد فایل‌های شافه‌ی جاری و زیر شافه‌های آن را که شامل پارامتر ورودی اسکرپت هستند، گزارش دهد. به عبارت دیگر، فایل‌هایی باید شمرده شوند که پارامتر ورودی در آنها رخ می‌دهد.

فایل `oslabe3.c` را به شکلی تغییر دهید که یک پردازش بسازد و تابع `compute1()` در پردازش جدید (همزمان با تابع `compute2()` در پردازش اصلی) اجرا شود. همچنین تابع `compute3()` باید (پس از پایان فراخوانی دو تابع `compute1()` و `compute2()`) در پردازش اصلی اجرا شود.

## گروه چهارم

اسکرپت پشته‌ای بنویسید که از تمامی فایل‌های شافیه جاری که پسوند «.txt» دارند، نسخه‌ی پشتیبان (Backup) بگیرد. نسخه‌ی پشتیبان فایل به الگوی abc.txt باید فایل با نام abc\_1.txt باشد.

در فایل oslab4.c یک پردازش ساخته می‌شود و تابع `compute1()` در آن فراخوانی می‌شود. پردازش اصلی نیز تابع `compute2()` را فراخوانی می‌کند و خروجی این تابع را چاپ می‌کند. این فایل را به صورتی تغییر دهید که خروجی تابع `compute1()` با استفاده از لوله از پردازش جدید به پردازش اصلی انتقال یابد و پردازش اصلی مجموع مقدار برگشت داده شده از دو تابع را چاپ نماید.

## گروه پنجم

اسکرپت پشته‌ای بنویسید که نام تمامی فایل‌های شافیه جاری و زیر شافیه‌های آن را چاپ کند که خط تکراری ندارند.

در فایل oslab5.c یک ریسمان جدید ساخته می‌شود و تابع `compute1()` در آن فراخوانی می‌گردد. ریسمان اصلی تابع `compute2()` را فراخوانی می‌کند و خروجی این تابع را چاپ می‌کند. این فایل را به صورتی تغییر دهید که خروجی تابع `compute1()` به ریسمان اصلی انتقال یابد. سپس، ریسمان اصلی مجموع مقدار برگشت داده شده از دو تابع را چاپ نماید.

## گروه ششم

اسکرپت پشته‌ای بنویسید که به ازای هر فایل در شافیه جاری و زیر شافیه‌های آن، تعداد رخداد عبارتی که به عنوان پارامتر اول اسکرپت داده می‌شود را چاپ کند.

در فایل oslab6.c سه ریسمان جدید ساخته می‌شوند و هر یک از این سه ریسمان مقدار متغیر سراسری `first` را تغییر می‌دهند. این فایل را به صورتی تغییر دهید که پس از تغییر متغیر `first` توسط اولین ریسمان، دو ریسمان دیگر این متغیر را تغییر ندهند. در این تغییر باید از دسترسی‌های همزمان به متغیر `first` جلوگیری کنید.

## گروه هفتم

اسکرپت پشته‌ای بنویسید که نام همه‌ی فایل‌های شافه‌ی جاری با پسوند `.h` را چاپ کند که فایلی با همان نام ولی با پسوند `.c` در این شافه موجود نباشد.

در فایل `oslabe7.c` دو تابع فراخوانی می‌شوند. این فایل را به شکلی تغییر دهید که هر یک از این دو تابع در یک پردازشی جدید فراخوانی شود با این شرط که اجرای تابع `compute2()` در پردازشی جدید دوم پس از فتمه‌ی فراخوانی `compute1()` در پردازشی جدید اول شروع شود.