

آزمون‌های نهایی

در این بخش برخی سؤالات پیشنهادی برای ارزشیابی نهایی درس آزمایشگاه سیستم عامل ارائه می‌شوند.

توازی ریسمانی

در تمرین هفتم، برنامه‌ی داده شده به دو قسمت شکسته شد و قسمت اول در یک ریسمان و قسمت دوم در ریسمان دیگری اجرا گشت. در این تمرین نیز از ریسمان‌ها برای همروندی اجرا می‌شود. در فایل داده شده ده بار تابع `proc()` صدا زده می‌شود. شما باید این فایل را تغییر دهید: ده ریسمان ایجاد نمایید و در هر یک از این ریسمان‌ها، یک بار تابع `proc()` را با ورودی مناسب فراخوانی نمایید (در ریسمان `i`-ام ورودی `i` را به تابع `proc()` بدهید). بدیهی است که در ریسمان اصلی، پس از ایجاد ریسمان‌ها باید منتظر خاتمه‌ی همه‌ی آنها بمانید؛ برای این کار می‌توانید شناسه‌های ریسمان‌ها را (ورودی اول تابع `pthread_create()`) در یک آرایه نگه دارید.

فایل‌های مشابه

اسکرینیتی بنویسید که با گرفتن آدرس یک فایل (پارامتر اول) و آدرس یک شاخه (پارامتر دوم)، آدرس فایل‌هایی که در شاخه‌ی مشخص شده (و زیر شاخه‌های آن) قرار دارند و محتوای آنها با محتوای فایل مشخص شده برابر است را چاپ کند. برای مقایسه‌ی محتوای فایل‌ها می‌توانید از دستور «cmp» استفاده نمایید. این دستور آدرس دو فایل را دریافت می‌کند و تنها در صورتی که محتوای دو فایل برابر باشند، با موفقیت خاتمه می‌یابد (که با کد برگشتی صفر مشخص می‌شود).

ارتباط دوطرفه

در تمرین ششم برنامه‌ی داده شده به دو پردازش شکسته شد و با استفاده از یک لوله داده‌های مورد نیاز پردازش دوم به آن منتقل گشت. در این تمرین باید از دو لوله استفاده نمایید: یک لوله برای انتقال ورودی و یک لوله برای دریافت خروجی. بنابراین تمرین ششم را تکرار کنید و خروجی `cons()` را در هر بار اجرا از راه لوله‌ی دوم به پردازش اصلی انتقال دهید. در پردازش اصلی نیز پس از فرستادن خروجی تابع `prod()` به لوله، مقدار برگشت داده شده توسط پردازش فرزند را از لوله‌ی دوم بخوانید و چاپ نمایید.

بزرگ‌ترین عدد

در تمرین هفتم، برنامه‌ی داده شده به دو قسمت شکسته شد و قسمت اول در یک ریسمان و قسمت دوم در ریسمان دیگری اجرا گشت. در این تمرین به صورت دیگری یک برنامه به دو قسمت شکسته می‌شود. در تابع `main()` فایل داده شده فایل `calc()` بیست بار با ورودی‌های متفاوت فراخوانی می‌شود و بزرگ‌ترین مقدار برگشت داده شده از این تابع چاپ می‌گردد. در این تمرین باید یک ریسمان ایجاد شود و ده بار از این فراخوانی‌ها به این ریسمان انتقال یابند. برای یافتن بزرگ‌ترین مقدار ممکن، دو ریسمان می‌توانند از یک متغیر `Global` استفاده نمایند ولی باید دسترسی‌های همزمان با استفاده از یک قفل `Mutex` مدیریت شوند.

تابع خطرناک

با بررسی کد یک برنامه متوجه شده‌اید که فراخوانی‌های تابعی مشکل امنیتی ایجاد می‌کنند. در این تمرین، اسکریپتی بنویسید که با گرفتن نام تابع خطرناک (پارامتر اول) و آدرس شاخه‌ی حاوی برنامه‌ی مورد نظر (پارامتر دوم) عملیات زیر را انجام دهد: فهرست فایل‌هایی که شامل آن عبارت هستند را چاپ نماید. در این فایل‌ها، به نام تابع پیشوند «warn_» اضافه نماید.