گزارش تمرین عملی دوم

ریحانه شاهوفیان ۹۹۵۲۱۳۶۱

۱) در ورودی ۲ عدد n و m را می‌گیریم و آرایه ای به سایز n. می دانیم n بر m بخش پذیر است. در قسمت اول این سوال m-1 بار fork() صدا زده می‌شود که m process ایجاد می‌شود که این process ها هرکدام یک بخش از آرایه را sort می‌کنند. بعد از آن process های child را kill می‌کنیم و با process های (parent) کار merge این m بخش را انجام می‌دهیم. و در نهایت آرایه‌ی sort شده را چاپ می‌کنیم و به عنوان خروجی می‌دهیم. در قسمت دوم این سوال تمامی کارها مثل قسمت اول است فقط نداریم و بجای آن m thread داریم که هرکدام sort را انجام می‌دهند ولی می‌دانیم thread ها به صورت موازی انجام می‌شوند در کد باید حواسمان باشد تا بخشی از آرایه را برای sort از دست ندهیم یا بخشی را بیشتر از یک بار sort نکنیم که برای حل این مشکل از pthread_join استفاده می‌کنیم.

2) می دانیم در حالت عادی وقتی client درخواست ثبت می‌کند process درخواست را اجرا می‌کند و پس از اتمام دوباره منتظر درخواست جدید می‌ماند. در قسمت اول این سوال با استفاده از fork() یک پراسس جدید ایجاد می‌کنیم تا یکی از پراسس ها به کلاینت رسیدگی کند و دیگری منتظر کلاینت جدید شود. در قسمت دوم این سوال هر درخواست client با thread انجام می‌شود که باعث می‌شود client منتظر تمام شدن درخواست client قبلی نباشد. نکته‌ای که در کد وجود دارد این است که برای پاس دادن آرگومان‌های تابع void* از struct استفاده می‌کنیم.

2-3) تفاوت های multi threading و multi processing :

این ۲ روشها برای multi tasking هستند که به ما این امکان را می‌دهند که برنامه‌ها را به صورت parallel (موازی) اجرا کنیم. در روش multi-processing بیشتر از یک processor داریم که به صورت همزمان فرآیند پردازش را انجام می‌دهند. در این روش CPU و cache به صورت مجزا هستند و memory به صورت مشترک به انجام پردازش می‌پردازد و چون تعداد CPUها را اضافه کرده باعث افزایش سرعت در محاسبات می‌شود. در روش multi-threading چند thread به صورت همزمان به اجرا می‌پردازند و چون threadها در یک process مشترک ایجاد می‌شوند، همه‌ی آن‌ها یک CPU مشترک دارند که باعث سرعت پردازش بالا می‌شود.

در روش multi processing نسبت به multi_threading میزان مصرف و فشار از CPU بیشتر است.

در روش multi-processing هر process یک فضای آدرس مجزا دارد ولی در multi-threading فضای آدرس بین threadها مشترک است.

3) فایلی به نام MyText.txt در directory که کد C وجود دارد هست. در قسمت اول سوال با fork و exec کدی را پیاده سازی کردم که به هنگام اجرا آرگومان اول کد را به عنوان اسم فایل یعنی MyText.txt می‌گیرد و با کد مقابل در Terminal محتویات فایل در cmd چاپ می‌شود. ./HW2-3-1 MyText.txt

در قسمت دوم همه چیز مثل قسمت قبل است فقط بجای fork و exec از system() استفاده می‌کنیم که به عنوان ورودی آن command مورد نظر یعنی cat MyText.txt داده می‌شود و در Terminal محتویات فایل چاپ می‌شود