

گزارش کار تمرین عملی دوم – محمد اصولیان 99521073

Q1

Q1_1

وارد پوشه q1_1 شوید. دستور “make -f Makefile” را وارد کنید تا برنامه اجرا شود. ابتدا عدد n و سپس عدد m از شما گرفته می‌شود، سپس n عدد رندوم جنریت و نمایش داده می‌شود و مراحل سورت کردن و آرایه نهایی چاپ می‌شود.

Q1_2

نحوه اجرا دقیقا مانند q1_1

Q2

Q2_1

وارد پوشه q2_1 شوید. با دستور “make -f Makefile server” سرور ران می‌شود. و با دستور “make -f Makefile clean” فایل آبجکت ساخته شده حذف می‌شود.

پورتی که سرور روی آن اجرا می‌شود ابتدای برنامه نمایش داده شده. با باز کردن مرورگر و وارد کردن چنین درخواست پشت سر هم میبینیم که سرور به همه درخواست ها همزمان رسیدگی میکند.

Q2_2

نحوه اجرا دقیقا مانند q2_1

Q2_3

در multi processing، چندین پروسس همزمان با هم مشغول هستند. هر پروسس سهم خود را از CPU و Memory خواهد داشت. بنابراین در multi processing استفاده بیشتری از CPU میشود و این کار سرعت بیشتری به انجام تسک ها میبخشد. از طرفی ساختن یک پروسس در multi processing کاری زمان بر است و به راحتی thread ساخته نمیشود.

در multi threading چندین ترد در یک پروسس همزمان مشغول هستند. با توجه به این که تنها یک پروسس داریم، مجموع سهم ترد ها از CPU تفاوتی با یک ترد نمیکند. تنها جابه‌جایی تسک هاست که باعث میشود در multi threading سرعت بیشتری داشته باشیم. در multi threading بر خلاف multi processing که هر پروسس حافظه مخصوص به خود را داشت، همه ترد ها به یک حافظه مشترک دسترسی دارند و به همین جهت کنترل تداخل ترد ها در حافظه نیز باید انجام شود. از طرفی ساختن ترد خیلی ساده تر از ساختن پروسس است و زمان کم تری نیاز دارد.

به طور کلی thread ها light weight تر از process ها هستند.

Q3

Q3_1

وارد پوشه q3_1 شوید. دستور “make -f Makefile” را وارد کنید تا برنامه اجرا شود. در برنامه اجرا شده محتویات فایل doc.txt توسط fork و execvp نوشته شده در فایل read_file.c نمایش داده می‌شوند.

Q3_2

نحوه اجرا دقیقا مانند q3_1

Q3_3

دستور system در واقع یک رابط بین process و operating system است. این دستور میتواند سرویس های OS را از طریق یک API برای برنامه‌نویس فراهم کند. تابع system در c یک کامند دریافت میکند و این کامند را اجرا میکند. System برای اجرای این کامند، یک پروسس جدید ایجاد میکند و بر خلاف دستور Exec که image پروسس اجرایی را جایگزین میکند، image پروسس در حال اجرا را تغییر نمیدهد. به همین علت هنگام استفاده نیازی به فورک کردن ندارد.

به طور خلاصه system طراحی شده تا یک interface بین OS و process ها باشد.