به نام ایزد یکتا

آزمایش ششم درس آزمایشگاه سیستم عامل

استاد: مهندس قاسمی

تهیه کننده: بردیا اردکانیان

۹۸۳۱۰۷۲

آزمایش اول)

برای این بخش، ابتدا برای ایجاد پردازه‌های reader و writer از fork می‌خواهیم استفاده کنیم. که این پردازه‌ها همگی به یک memory shared دسترسی دارند پس یک shared memory ایجاد می‌کنیم و مشخص می‌کنیم که روی آن می‌خواهیم از mutex استفاده کنیم. حال mutexچیست؟ mutex مانند یک قفل است که برای تردها استفاده می‌شود، وقتی یک ترد آن را قفل می‌کند و وارد Critical Section می‌شود، ترد دیگری نمی‌تواند وارد آن قسمت شود و آن دیتا را تغییر دهد تا زمانی که قفل توسط همان ترد قبلی باز شود.

در اینجا چون از fork استفاده کردیم و میخواهیم از مکانیزم mutex نیز استفاده کنیم، shared memory را به صورت آرایه ای از استراکت‌ها تعریف می‌کنیم که در آن مقدار counter ،تعداد پردازه‌های reader و mutex را نگه داری می‌کنیم.

در واقع reader همان mutex up و writer به عبارتی mutex down است.

حال بعد از ایجادreaders ها و writer ،به این صورت عمل می‌کنیم که هنگامی‌که writer بخواهد بنویسد mutex را قفل می‌کند و پس از انجام کار آن را آزاد می‌کند. و در readers ،اولین خواننده mutexرا قفل می‌کند و آخری آزاد می‌کند و به همین دلیل است که به تعداد readers نیاز داریم و آن را داخل استراکت تعریف می‌کنیم.

آزمایش دوم)

در این بخش، مثل بخش قبلی از mutex استفاده می‌کنیم به این صورت که ۵ ترد می‌سازیم که هر کدام نماینده یک فیلسوف است. حال برای هر چاپ استیک می‌خواهیم یک قفل قرار دهیم. برای اینکار از pthread_mutex_t استفاده می‌کنیم و یک آرایه ۵ تایی از آن می‌سازیم. در هر ترد کاری که می‌کنیم این است که آن فیلسوف، چاپ استیک‌های دو طرف خودش را بر میدارد وآن‌ها را قفل می‌کند تا بقیه از آن‌ها نتوانند استفاده کنند. سپس غذا می‌خورد و پس از خوردن قفل را آزاد می‌کند .درواقع pthread_mutex_t کاری که می‌کند این است که آن ترد با قفل کردن اجازه نمی‌دهد تردهای دیگر به آن دسترسی پیدا کنند.

همچنین می‌توانیم از مکانیزم semaphore برای قفل کردن استفاده کنیم.