گزارش پروژه درس سیستم عامل

سید ابوالفضل قاسم زاده 400243064

در این پروژه در کلاس اصلی آن، یعنی ConcurrentQueue موارد لازم برای پیاده سازی در آن انجام گرفته است، این موارد که بصورت کلی توسط 3 عدد سمافور و یک شمارنده(counter) پیاده سازی شده است و شرح دقیق آن بدین شرح است :

1. queue : بیانگر همان صف اصلی این کلاس است که تمامی داده های تولیدی در آن ذخیره میشود و با استفاده از ساختمان داده‌ی لیست پیوندی پیاده سازی شده است. جنس دیتای آن از کلاس Message میباشد.
2. producer\_lock : سمافوری برای موقعی که یک پیام میخواهد در صف نوشته شود. مقدار اولیه آن 1 بوده چون در ابتدا حداقل یک تولید کننده میتواند پیام تولید کند و در همان تابع send\_msg مقدار آن تغییر و acquire یا release میشود.
3. consumer\_lock2 : سمافوری برای مواقعی که چندین مصرف کننده با هم و همزمان خواهان مصرف یک پیام از صف هستند استفاده میشود. هنگامی که یک ترد وارد شد، acquire میشود تا ترد دیگری وارد نشود، سپس هنگامی که پیام خود را دریافت کرد و کارش تمام شد، release میشود.
4. consumer\_lock1: سمافوری برای موقعی که یک مصرف کننده خواهان مصرف یک پیام از صف هست اما پیامی در صف موجود نیست، استفاده میشود. هنگامی که یک تولید کننده پیامی را داخل صف نوشت، این سمافور را release میکند که این یعنی یک مصرف کننده میتواند پیامی بخواند(شمارنده thread\_counter نیز برای همین زیاد میشود تا مقادیر مجاز ترد هایی که میتوانند وارد شده و پیامی را بخوانند را بشمرد.) سپس اگر پیامی در صف نبود، یعنی مقدار thread\_counter صفر بوده و یعنی تعداد ترد های مجاز صفر است، acquire شده تا وقتیکه یک تولید کننده پیامی در صف بنویسد.
5. thread\_counter : همانطور که گفته شد، مقدار ترد های مجاز برای دریافت پیام را مشخص میکند. مقدار آن در هنگام نوشتن پیام افزایش و هنگام خواندن پیام کاهش میابد.