

۱- ۱. قفل mutex: ویژگی انحصار متقابل، دارای دو تابع lock یا acquire و unlock یا release است.

۲. قفل semaphore: دارای دو تابع wait و signal.

۳. قفل monitor به صورت کلاس است. دارای ویژگی انحصار متقابل و متغیرهای condition است، اگر شی ای که از جنس مانیتور است تعریف کنیم اگر یکی از توابع یکی از این شی ها توسط thread ای در حال استفاده باشد در آن لحظه thread دیگری نمیتواند از آن ها استفاده کند. دو تابع wait که در هر صورت بند منتظر میشود و signal یا notify که یکی از بندها را از حالت انتظار خارج میکند، دارد و همچنین notify_all دارد که همه بندهای در حال انتظار را از حالت انتظار خارج میکند. همه این قفل ها از دسترسی همزمان به متغیرهای مشترک و اشتباه در اطلاعات کد جلوگیری میکنند.

۲- باقرار دادن متغیر قفل از نوع سمافور با مقدار اولیه N، با فراخوانی تابع wait، N بند میتواند وارد شده و دسترسی داشته باشند و بعد از آن متغیر قفل صفر میشود که در نتیجه تاوقتی که یکی از بند ها باتمام شدن خود signal را که به متغیر قفل یکی اضافه میکند صدا بزند، بند های N+1 به بعد منتظر میشوند.

۳- با توجه به wait(sems[i]) با حداقل ۹ بند بن بست رخ میدهد اگر همه خط ۶ را فراخوانی کنند همه باید برای فراخوانی خط ۷ منتظر بمانند.

۴- اگر بند یک یا دو زودتر اجرا شوند، بند یک در خط ۶ منتظر می شود، بند ۲ مقدار val را ۳ میکند و سپس قفل را آزاد میکند در این حالت اگر بند ۲ زودتر ادامه کار را انجام دهد، در خط ۱۴ دوباره قفل را فعال کرده و مقدار را ۴ میکند، در غیر این صورت بند ۱ زودتر ادامه دهد قفل را فعال کرده مقدار val به ۹ تغییر کرده و بند ۲ در خط

۱۴ منتظر میماند سپس بند ۱ قفل را برداشته و بند ۲ مقدار را ۱۰ میکند. پس مقدار val می تواند ۴ یا ۱۰ شود.