امیر محمّد آقاپور ۹۶۳۱۰۰۸ تمرین ۷

۱. برنامه کامپایل نمی شود زیرا count متغیّر محلی تابع interleave است و بین thread ها مشترک نیست و توسط تابع test هم قابل دسترسی نمی باشد. اگر آن را تبدیل به متغیر گلوبال کنیم(خارج از تابع تعریف شود) آن گاه مقدار نهایی می تواند در بازه MAX تا $2 \times MAX$ قرار گیرد. MAX در حالتی است که ابتدا یک ترد مقدار 0 را برای count بخواند ولی بعد از افزایش یک واحد، حاصل را در رم ننویس، امّا ترد دیگر تا پایان بشمارد و سپس ترد اول بشمارد. $2MAX$ هم وقتی هر یک در هر گام بخواند و یکی زیاد کرده و بنویسد، بدون مداخله دیگری در میان این اعمال.

۲.

انحصار متقابل: ندارد. حالتی که نقض می شود:

$$process\ 0\ enters\ region \rightarrow interested[0] = True \rightarrow turn = 1$$
$$\rightarrow process\ 2\ enters\ region \rightarrow interested[2] = True \rightarrow turn = 0$$
$$\rightarrow process\ 2\ busy\ waiting \rightarrow process\ 0\ begins\ executing$$
$$\rightarrow process\ 1\ enters\ region \rightarrow interested[1] = True \rightarrow turn = 2$$
$$\rightarrow process\ 1\ busy\ waiting \rightarrow process\ 2\ begins\ executing$$

همان طور که مشاهده می شود همزمان دو فرآیند ۰ و ۲ در حال اجرای ناحیه ی بحرانی هستند.

پیشروی: ندارد. حالتی که نقض می شود:

$$process\ 0\ enters\ region \rightarrow interested[0] = True \rightarrow turn = 1$$
$$\rightarrow process\ 1\ enters\ region \rightarrow intrested[1] = True \rightarrow turn = 2$$
$$\rightarrow process\ 1\ busy\ wainting \rightarrow process\ 0\ busy\ wainting$$

همان طور که مشاهده می شود دو فرآیند ۰ و ۱ در حال انتظار هستند در حالی که هیچ فرآیندی ناحیه بحرانی را اجرا نمی کند. توجه می کنیم حتی اگر فرآیند ۳ بیاید و نوبت را به فرآیند ۰ داده و ناحیه بحرانی اش را اجرا کند، همچنان چون در پایان اجرا نوبت به فرآیند دیگری منتقل نشده، دو فرآیند دیگر همچنان منتظر خواهند ماند.

انتظار محدود: دارد. زیرا حداکثر ۲ فرآیند قبل از فرآیند درخواست دهنده وارد ناحیه بحرانی می شوند(با این فرض که پیشرفت رخ می دهد که یعنی فرآیند ها در فاصله زمان مشخص به ناحیه بحرانی وارد می شوند).

۳.

الگوریتم هر سه شرط انحصار متقابل و پیشرفت و انتظار محدود را دارد.

متغیر k اندیس فرآیندی که باید ناحیه بحرانی را اجرا کند نگه میدرد و متغیر control وضعیت هر فرآیند را که دارای یکی از سه مقدار صفر و یک و دو است که صفر برای حالت بدون تأثیر(idle)(وقتی خارج از ناحیه بحرانی است و درخواست ورود هم نداده است)، ۱ برای منتظر(وقتی منتظر ورود به ناحیه است) و ۲ وقتی داخل ناحیه بحرانی شده و در حال اجرای کد مربوطه است.

الگوریتم به طور کلی به این صورت عمل میکند:

۱. هر فرآیند برای ورود به ناحیه بحرانی، در مرحله اوّل بعد از تغییر وضعیت خود به منتظر، در صورتی که فرآیند دیگری منتظر یا در حال اجرای کد بحرانی است؛ در همین مرحله می ماند.

۲. سپس به صورت موقت وضعیت خود را به در حال اجرای ناحیه بحرانی تغییر می دهد و سپس بررسی میکند که آیا فرآیند دیگری در این وضعیت وجود دارد یا خیر. در صورت وجود به مرحله ۱ بر میگردد.

۳. بررسی میکند که فرآیندی که در حال حاضر دارای نوبت(متغیر k) است در حال اجرا نباشد که در غیر این صورت باز به مرحله ۱ بر میگردد. در این صورت مطمئن از انحصار متقابل می شویم.

۴. نوبت را میگیرد و کد بحرانی را اجرا میکند.

۵. پس از اجرای کد ناحیه بحرانی، نوبت را به فرآیندی که دارای وضعیت بی تأثیر نیست(منتظر یا در حال اجرای ناحیه بحرانی) می دهیم و وضعیت خود را به بدون تأثیر تغییر می دهیم. در این صورت از انتظار محدود مطمئن می شویم. همچنین اگر فرآیند با این شرایط پیدا نشود، خودمان نوبت را نگه میداریم.