به نام خدای رنگین کمان

امیرعادل گودرزی کلالی 9931099 تمرین تحویلی سوم درس سیستم های عامل

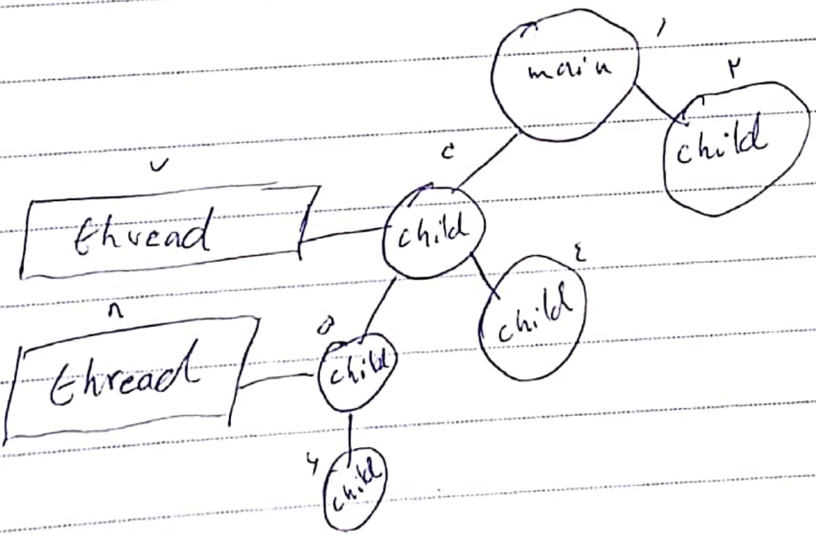
1) الف) ready، اینست برنامه‌ای است که روی رم قرار گرفته و منتظر است تا نوبت cpu رفته و اجرا گردد. اما waiting وضعیتی از برنامه است که در حالت اجرا بوده اما به این دلیل که منتظر است یک event انجام شود به حالت waiting می‌رود.

ب) انتشار وقفه یا انتظار برای دستگاه I/O

2) برای انجام شدن multitasking و اینکه چندین فرایند بتوانند در یک بازه زمانی مشخص، پیشرفت داشته باشند، cpu به فرایندها محدودیت‌های زمانی می‌دهد.

با کمتر بودن زمان باعث می‌شویم که هر فرایند پیشرفت کمتری داشته باشد و از طرفی نیز باعث بالا بردن تعداد context switch ها شده و overhead بیشتری تولید کنیم که یعنی کار غیرمفید بیشتر. با زیادتر کردن زمان، شاید باعث شویم یک فرایند، پیشرفت بیشتری داشته باشیم اما دیگر فرایندها باید بیشتر صبر کنند و پیشرفت همزمان فرایندها را کمتر مشاهده خواهیم کرد.

3)



threads → 8

Process → 6

Subject:

Date:

(الف) ④

```
flag[i] = true;
turn = i;
while(flag[j] && turn = = i);
        cs
flag[i] = false;
```

```
flag[j] = true;
turn = j;
while(flag[j] && turn = = j);
        cs
flag[j] = false;
```

با توجه به درک خودم از کد حتی اگر flag ها نیز، هردو true باشند با توجه به اینکه، مقدار turn قبل از هر while مقدار دهی می شود و فقط می تواند یکی از دو مقدار باشد فقط یکی از دو فرایند زیان می تواند وقفه گذرا پس انتظار متقابل داریم.

حال درستی ست چپ با اجرای ④ شرط while false شده و cs اجرای شود پس پیشرفت داریم.

از آنجایی که این الگوریتم به بیشترین شباهت داشته و همچنین اگر هر ④,③,②,① اجرای درستی ها جابه شده و cs سمت راست اجرای گردد پس می توان عدالت و انتظار محدود این الگوریتم را نیز تایید کرد.

(ب)

```
flag[i] = true;
turn = i
while(flag[j] && turn = = i);
        sc
flag[i] = false
```

```
flag[j] = true;
turn = j
while(flag[i] && turn = = j);
        sc
flag[j] = false
```

Subject:

Date:

در این مورد نیز مانند قبلی حتی اگر هر دو فرایند true باشند چون که turn فقط یک مقدار دارد و اگر یکی شرط while را دارد قطعاً یکی بلاک شده و دیگری sc اش را اجرا می‌کند. پس انحصار متقابل و پیشرفت را داریم.

همچنین می‌توان گفت عدالت و انتظار محدود نیز رعایت شده است، زیرا که هرگاه که فرایندی شروع می‌شود مقدار turn را به خودش می‌دهد و شرط برابری turn با خودش را در while ای می‌گذارد که برای اجرا دیگر فرایند ها صبر می‌کند.

ج

از آنجایی که race condition رخ نمی‌دهد یعنی عملیاتی که موجب ایجاد این شرط می‌شود به صورت سری انجام می‌گیرد. همچنین با در نظر گرفتن که های مصرف کننده و تولید کننده که در حالات خالی یا پر بودن بافر عملیاتی انجام نمی‌دهند می‌توان تضمین کرد که موارد الف و ب رخ نمی‌دهد.