

بسمه تعالی



دانشکده مهندسی کامپیوتر

دکتر انتظاری

سیستم‌های عامل

پاسخنامه تمرین سری اول

آرمان حیدری – غزل زمانی نژاد

## 1. پاسخ سوال اول

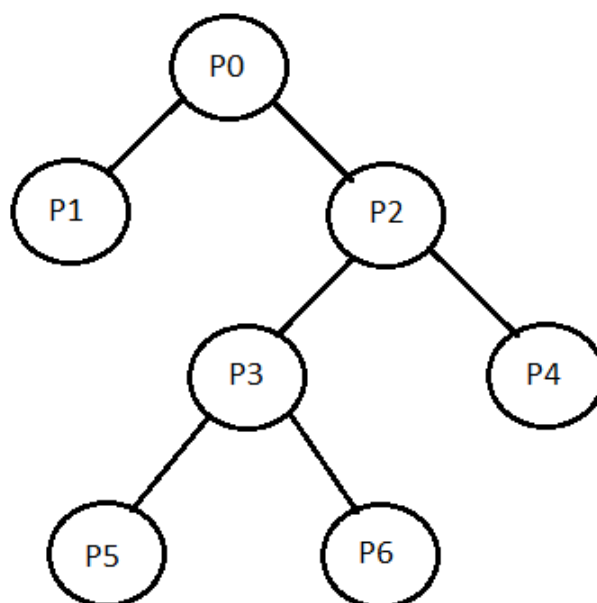
**الف)** به طور میانگین shared memory سرعت بیشتری دارد. زیرا kernel در روش message passing مجبور به دخالت است و باید پیام ها را لینک کند و ... در حالی که در shared memory بدون واسطه و بدون اتلاف وقت برای ارسال و دریافت پیام ها کار انجام می شود.

**ب)** در روش shared memory ، چون باید دیتا ها cache شوند و همین فرآیند ممکن است باعث ایراداتی در به روز بودن دیتا ها شود که گاهی مشکل ساز خواهد بود. اما چنین مشکلاتی در message passing نیست زیرا هر process حافظه و data های خودش را دارد و با ارسال پیام ها و دریافتشان تغییرات مربوطه انجام می شود.

## 2. پاسخ سوال دوم

می دانیم در زبان C، اگر شرط عددی غیر صفر باشد، true در نظر گرفته می شود. و اگر صفر باشد false خواهد بود. همچنین می دانیم که دستور fork برای child process ، 0 را بر میگرداند و برای parent process ، عددی مثبت را.

درخت process های این سوال با فرض شروع اجرای برنامه توسط P0 به این شکل خواهد بود:



چون شرط اولین if ، and دارد پس اگر اولین fork خروجی صفر بدهد (در child باشیم)، یعنی p1 برنامه مستقیم سراغ printf می رود و تمام می شود. ولی اگر اولین fork خروجی عدد بدهد (در p0 باشیم) fork دوم اجرا می شود و p2 ساخته می شود که فرزند p0 است. چون این بار ! داریم، پس فقط برنامه در p2 وارد می شود و در p0 مستقیم به printf می رود.

حالا شرط دوم که or دارد، وقتی اولین fork عدد مثبت باشد یعنی در p2 باشیم پس p3 ساخته شده و وارد شرط می شویم (بدون نیاز به اجرای fork بعدی چون شرط "یا" است و اولین آن عددی مثبت است). داخل شرط هم که fork اجرا می شود و لذا p4 ساخته می شود. در اینجا p4 و p2 تمام می شوند و هر دو printf را اجرا می کنند. و اما p3 باید fork دوم شرط را اجرا کند. اگر آن هم صفر باشد (یعنی در printf ) p5 اجرا میشود و تمام می شود. اما اگر صفر نباشد (در p3 ) دستور fork آخر اجرا می شود و p6 ساخته می شود. بعد هم p6 و p3 ، printf را اجرا می کنند و تمام می شوند. پس 7 تا process داشتیم که همگی یک بار دو را می نویسند و خروجی 2 2 2 2 2 2 2 خواهد بود.

### 3. پاسخ سوال سوم

- مدیریت دسترسی پردازه های مختلف اگر بخواهند به داده های مشترکی دسترسی داشته باشند، که هر پردازه به چه بخشی دسترسی داشته باشد یا اگر تغییری توسط یکی انجام شد در بقیه آپدیت شود و ... از چالش های پردازه های همزمان است.
- برقراری ارتباط بین پردازه های همزمان توسط سیستم عامل
- مشخص کردن بخشی هایی از برنامه که میتوانند همزمان انجام شوند و تفکیک آن ها از بخش هایی که باید به صورت متوالی انجام شوند. همچنین پیاده سازی برنامه ها به صورتی که حداکثر استفاده از همزمانی را ببریم نیز یکی از چالش هاست.

\*برای این سوال صرفا همین موارد وجود ندارد. میتوانید موارد دیگری هم ذکر کنید و اگر منطقی باشند نمره به آن‌ها تعلق میگیرد.

#### 4. پاسخ سوال چهارم

خروجی همچنان عدد 5 خواهد بود. چون پردازش فرزند، صرفا یک کپی از value را تغییر میدهد و وقتی در پردازش والد هستیم همچنان 5 است. در واقع حافظه مربوط به این دو پردازش مجزا است.

#### 5. پاسخ سوال پنجم

آ) Interrupt سیگنالی است که توسط device به cpu ارسال می‌شود. وقتی سیگنال دریافت شد، cpu کار خودش را متوقف کرده و execution را به مکان مناسب منتقل می‌کند.

ب) trap یا همان exception گونه‌ای از interrupt است که توسط نرم‌افزار تولید می‌شود. دلیل ایجاد شدن آن:

(۱) به دلیل ارور مثلا تقسیم بر صفر یا دسترسی نامعتبر به مموری

(۲) یک ریکوئست برای گرفتن سرویسی از سیستم عامل (systemcall)

ج) برنامه‌های کاربر می‌توانند باعث ایجاد trap بشوند. بعضی زبان‌های سطح بالا برای کاربر امکان ایجاد trap را فراهم می‌کنند (به عنوان مثال exception در جاوا). هدف از ایجادشان آن است که به cpu اعلام کنند که در نرم‌افزار یک اتفاق غیرمنتظره رخ داده یا کاربر سرویس خاصی را درخواست کرده است.

## 6. پاسخ سوال ششم

در multiprocessing، چند برنامه به صورت همزمان در حال اجرا هستند و باعث می‌شود cpu در تمامی زمان‌ها مشغول فعالیت باشد. Multiprogramming بهره‌وری cpu را افزایش می‌دهد و در عین حال به کاربر اجازه استفاده از چند برنامه را می‌دهد. به برنامه در حال اجرا process می‌گوییم.

در multitasking، cpu با سوییچ کردن بین پراسس‌ها به اجرای چند پراسس می‌پردازد. فرآیند سوییچ کردن اغلب اتفاق می‌افتد تا مقدار response time کم باشد. مثلاً زمانی که کاربر می‌خواهد از طریق I/O به سیستم ورودی بدهد، ممکن است سرعت او کم باشد. اگر از multitasking استفاده نشود در این زمان cpu، idle می‌ماند. اما در صورت استفاده از آن cpu می‌تواند بین سایر پراسس‌ها سوییچ کند.