سیستم عامل

تمرین سوم

علیرضا حیدری - ۹۴۳۱۰۵۷

۱.در بسیاری از اپلیکیشن ها نیازمندیم تا جواب هرچه سریعتر پیدا شود چون ممکن است ادامه اپلیکیشن منتظر نتیجه است و اگر در سرعت مناسب پاسخگو نباید برنامه دچار مشکل می شود. در چنین اپلیکیشن هایی نیازمند الگوریتم هایی به صورت B هستیم که جواب را هرچند که بهینه ترین حالت مهم نباشد اما در سرعت کافی میدهند. همچنین در حالتی که دیتاهای مشترک داریم نیز از الگوریتم B استفاده میکنیم

در حالاتی که از الگوریتم A استفاده میکنیم نیز اپلیکیشن هایی که زیاد مسئله پاسخگویی در زمان زیاد اهمیت خاصی ندارند قابل توجه است.

۲.تکنیک swapping :
این تکنیک به این صورت است که پراسس برای توقف یا جابجایی باید تعادلی در پراسس ها ایجاد کند. در این تکنیک اگر فرآیند ها زیاد باشد ممکن است به حذف منجر شود و در صورت کم بود پراسس ها به صف اجرا میروند.

تکنیک context switch :

این تکنیک به این صورت است که کرنل برای توقف یا جابجایی، پراسس را از cpu میگیرد و در pcb اش نگه میدارد.

۳.به صورت سه مرحله ی ذخیره ی حالت کنونی در pcb ، انتخاب فرآیندی برای اجرا از بین فرآیند های در انتظار و بارگذاری از pcb که هر سه نرم افزاری صورت میگیرد

۴.اگر سخت افزار چندین رجیستر داشته باشد نشانه گر میتواند به رجیستر جدید اشاره کند. اگر نیازی به ذخیره pcb نباشد بی استفاده است.
و در حالتی که هیچ برنامه ای در صف انتظار نباشد نیز به کار نمی آید.

۵.

۱.Parent and child don’t share all resources but execute concurrently
۲.Parent and child share all resources and execute concurrently
۳.parent and child share all resources and don’t execute concurrently
۴.parent and child don’t share all resources and don’t execute concurrently
درصورت قابل دسترس بودن برای هردو آنگاه هر دو توانایی تغییر را دارند و رابطه مستقیم است و نیازی به ارتباطات نیست اما این قابل تغییر دادن داده ها توسط هردو باعث می شود که در بعضی مواقع داده ها عوض شوند درصورتی که یکی از طرفین لازم داشته که داده دست نخورده باقی بماند
در صورتی که داده ها مشترک نباشند آنگاه میتوان داده ها را با ارتباط و message passing انتقال داد و سرعت اندکی پایین تر می آید.
در صورت انجام کارها به صورت هم روند آنگاه امکان وجود race condition زیاد می شود و انجام کارها مشکلی در برخواهد داشت. اما در صورتی که به صورت موازی و هم روند اجرا نشوند آنگاه مشکل race condition حل شده اما اجرای والد نیاز به توقف برای عملیات فرزند دارد و کارها نسبت به حالت هم روند سرعت کمتری دارد.

a.۶

این عبارت در تابع killhandler است که زمانی صدا زده می شود که تابع kill شده پس هیچگاه صدا زده نمیشود و هیچ گاه عبارت چاپ نمیشود

b

۰

۵

۱۰

۱۵

۲۰

.۷

۳ چاپ می شود.