

① - در وضعیت ready اگر به فرآیند PC حقوق برگردد اجرا شدن دارد و اجرای شود آن در وضعیت wait از آنجا که منتظر تمام شدن IO یا اتفاقی است. با دادن PC هم قابلیت اجرا ندارد.

2 - اگر process ای در حال اجرا باشد با آمدن interrupt که می تواند زمانی باشد (یا تاخیری که بشود اگر قرار باشد پیش تر از حد تعیین شده طول کشد) یا حتی برابری باشد، به وضعیت ready برود و PC از آن گزیده می شود.

فرآیند در حال اجرا اگر در وقفه از سمت IO / event آمد PC از فرآیند گزیده می شود و در حالت waiting می ماند تا وقتی که IO تمام شود و با اجرا شدن ISR مربوط به آن دستگاه به فرآیند خبر می دهد که می تواند دوباره شروع به کار کند در حالت ready می رود و PC در حال حفظ آن حقوق نمی گیرد.

← در وضعیت child termination می شود.

② ① دارد fork a child می شود و فرآیند child در وضعیت ready می شود.

② دارد fork a child می شود و فرآیند child (child خوانند) در وضعیت ready می شود.

③ دارد صف ready می شود.

④ دارد بخش درخواست IO و سپس صف IO می شود (مثلاً IO انجام شده و در وضعیت ready می شود).

⑤ دارد بخش wait for interrupt شده و ISR اجرای می شود و در وضعیت ready می شود.

⑥ دارد time slice expired می شود و PC از آن گزیده می شود و در وضعیت ready می شود.

③ به فوکل هر مقدار زمان اختصاص داده شده به فرآیند ها کمتر باشد، امکان بیشتری برای

اجرا شدن فرآیندهای بیشتری فراهم می شود که هر کدام زمان کمی برای اجرا دارند. این کار،

تعدادات interactive را بالا می برد. از طرفی برای تغییر فرآیندها، context switch

صورت می گیرد که بین خود سر بار زمانی دارد زیرا در این زمان هیچ کدام از فرآیندهای خواسته شده

توسط user اجرا نمی شود پس تا کم بودن زمان اختصاص داده شده به فرآیندها

سر بارهای زمانی زیادی در هر واحد زمان صورت می گیرد که این از بهره داری CPU کم می کند.

DATE / /

SUBJECT:

از طرف دیگری اگر زمان زیادی به هر فرآیند اختصاص داده شود، فرآیندی بیشتر اجرایی شود که زودتر در صف ready شده (first come first served, FCFS) و این بازدهی را کاهش می‌آورد زیرا شاید فرآیندی که آخر اضافه شده برای ما اهمیت اجرایی بالاتری داشته باشد که چون زمان اختصاص داده شده به فرآیندها زیاد است دیرتر به فرآیندهای آخر صف، نوبت اجرایی رسد.

با این توصیفات بهتر است به زمان خالی زیادی برای هر فرآیند در نظر گرفت (جدول‌گیری از FCFS) هم این زمان آنقدر کم باشد که سرور زمانی باعث کاهش بهره‌وری شود.

اگر سیستم‌های مدرن از زمان 10ms و زمان 10ms content switch (استفاده می‌کنند) بنابراین سرور زمانی نیست به توان خواندن کوچک دیگران چیست چینی است.