

بن بست پردازها (Process Deadlocks)

اگر مجموعه ای از پردازها در سیستم منتظر وقوع حادثه ای باشند که توسط دیگری انجام شود و هیچ کاری در سیستم پیش نرود، کوئیم سیستم دچار بن بست شده است.

شرایط وقوع بن بست: برای رخ دادن یک بن بست هر چهار شرط زیر باید برقرار باشند.

1- انحصار متقابل (Mutual Exclusion)

2- گرفتن و منتظر ماندن (Hold and wait)

3- عدم پس گرفتن (انحصاری بودن) (No preemption)

4- انتظار چرخشی (Circular Wait)

انحصار متقابل بدین معناست که منبع یا منابع در هر آن، فقط توسط یک پردازها قابل استفاده باشند.

گرفتن و منتظر ماندن یعنی پردازها یک سری منابع مورد نیازش را در اختیار گرفته، و منتظر منابعی است که در اختیار دیگر پردازها است عدم پس گرفتن بدین معناست که به اجبار نمی توان منابع یا منابعی را از پردازها پس گرفت.

انتظار چرخشی یعنی بایستی مجموعه ای از پردازشها $\{p_0, p_1, \dots, p_n\}$ وجود داشته باشد به طوری که p_0 منتظر منابعی از p_1, p_1 منتظر منبع از p_2 و $p_n \dots$ منتظر منابعی از p_0 باشد.

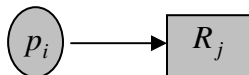
روش توصیف بن بست: استفاده از گراف تخصیص منابع (Resource Graph) $G(V, E) \leftarrow$

مستطیل $R =$ منابع R_1

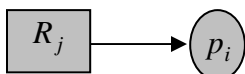
دایره $P =$ پردازها p_0

در گراف تخصیص منابع، گره ها از نوع منابع یا پردازها اند، که منابع با مستطیل نمایش داده می شوند و تعداد نمونه های آن با نقطه داخل آن مشخص می شوند و پردازها با دایره مشخص می شوند.

یال های گراف تخصیص منابع جهت دار می باشند که یا از پردازها به منابع می باشد (بدین معناست که پردازها p_i منتظر منبع R_j است) به این شکل.



یا از منابع به پردازها می باشد (بدین معناست که یک نمونه از منبع R_j در اختیار پردازها p_i می باشد)



به این شکل.

می توان اثبات کرد که اگر گراف دارای هیچ سیکلی (حلقه یا loop) نباشد، هیچ پردازشی در سیستم در بن بست نخواهد بود.

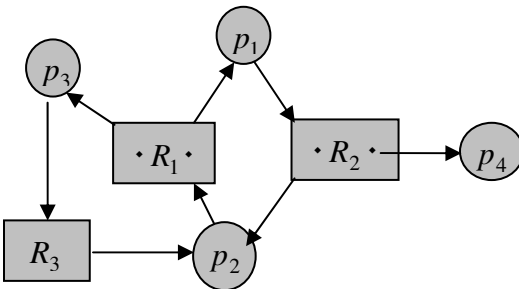
اگر گراف دارای سیکلی باشد، احتمال دارد بن بست وجود داشته باشد. پس وجود حلقه در گراف شرط لازم برای بن بست است و نه شرط کافی

□ اگر در گراف هر منبع دقیقاً یک نمونه داشته باشد، آنگاه اگر گراف حلقه داشته باشد، بدین معناست که تمام بن بست رخ داده است.

ولی اگر هر نوع منبع نمونه‌های متعددی داشته باشد، آنگاه حلقه الزاماً به معنای وقوع بن بست نیست

□ اگر در گراف حلقه‌ای وجود نداشته باشد، آنگاه سیستم در حالت بن بست نیست

مثال: آیا گراف زیر در بن بست قرار دارد.



حل: خیر

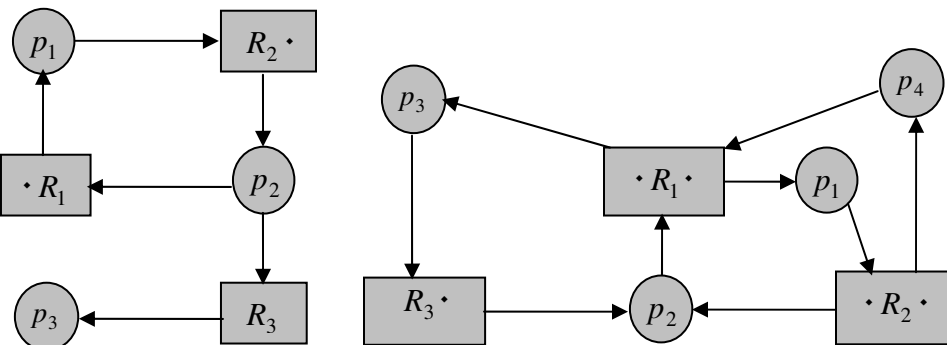
توضیح: در شکل گفته شده در گراف حلقه وجود دارد ولی سیستم (پردازه‌ها)

در بن بست نیستند، زیرا پردازه P_4 به منابع دیگری نیاز ندارد و بنابراین

اجرایش را به اتمام رسانده و منبع R_2 را، رها می‌سازد با رها ساختن منبع R_2 بن بست شکسته می‌شود. منبع R_2 به پردازه P_1 تفصیص داده می‌شود پردازه P_1 تمام منابع مورد نیازش را در اختیار دارد، اجرایش را به اتمام رسانده و منابع را آزاد می‌سازد. در این حال منبع R_1 را به پردازه P_2 داده، پردازه P_2 نیز با در اختیار داشتن تمام منابع مورد نیازش، اجرایش را به پایان رسانده و تمام منابع را آزاد می‌سازد. که یکی از منابع R_3 است. منبع R_3 به پردازه P_3 تفصیص داده شده و این پردازه با در اختیار داشتن منابع اجرایش را به پایان می‌رساند.

مثال: آیا گراف‌های زیر در بن بست قرار دارند

حل: بله در هر دو بن بست وجود دارد.



فهرایند استفاده از منبع:

1- در خواست منبع (Request): اگر پردازه‌ای به منبعی نیاز داشته باشد، دستور سیستمی درخواست منبع را می‌دهد، که اگر آن منبع آزاد باشد، به پردازه درخواست‌کننده تفصیص داده می‌شود، و گرنه پردازه می‌بایست منتظر منبع باشد

2- بکارگیری منبع (allocation): در صورت وجود منبع درخواست شده، منبع به پردازه درخواست‌کننده توسط سیستم تفصیص داده می‌شود.

3- آزاد کردن منبع (Release): پردازه پس از استفاده از منبع تفصیص یافته، منبع را به سیستم برمی‌رداند.

□ در خواست منبع و آزاد کردن آن، توسط دستورات سیستم عامل (System Call) انجام می‌شود. مثلاً برای فایل دستورات open

و close و برای سمافور wait و signal انجام می‌شود. و همواره سیستم کلیه درخواست‌ها تفصیص می‌دهد و آزاد شدن منابع را تحت نظر □ دارد