مهر زمانی خواندن و نوشتن:

برای هر داده Q مهر زمانی خواندن و نوشتن به صورت زیر است:

- W-TS(Q) : مهر زمانی نوشتن داده Q ،که برابر است با بزرگترین مهر زمانی تراکنشی که روی Q نوشته است.
- R-TS(Q) : مهر زمانی خواندن داده Q ،که برابر است با بزرگترین مهر زمانی تراکنشی که داده Q را خوانده است.

با اعمال قواعد خواندن و نوشتن،پروتکل های مبتنی بر مهر زمانی تضمین می کنند که دستورات r و w که با هم برخورد دارند به ترتیب مهر زمانی اجرا شوند و سریال پذیر باشند.

قواعد خواندن :

فرض کنید تراکنش Ti شامل یک دستور read(Q) است:

1) اگر TS(Ti)<=W-TS(Q) آنگاه تراکنش Ti داده ای را می خواهد که مقدارش بعدا نوشته می شود. پس در این حالت با دستور خواندن تراکنش موافقت نمی شود و ترامنش رد می شود.
2) اگرTS(Ti)>=W-TS(Q) آنگاه دستور خواندن تراکنش Ti اجرا می شود و مهر زمانی خواندن Q ،با ماکزیمم مهر زمانی تراکنش Ti ومهر زمانی خواندن Q مقداردهی می شود.

قواعد نوشتن :

فرض کنید تراکنش Ti شامل یک دستور write(Q) است:

1) اگر TS(Ti)<R-TS(Q) یا TS(Ti)<W-TS(Q) آنگاه با دستور نوشتن تراکنش موافقت نمی شود و تراکنش Ti رد می شود.
2) در غیر این صورت دستورنوشتن اجرا می شود و مهرزمانی نوشتن Q نیزبا TS(Ti) مقدار دهی می شود.

روش های مدیریت بن بست:

1) چشم پوشی (ignore)
2) فرصت (timeout)
3) پیشگیری(prevention)
4) اجتناب (avoidance)
5) تشخیص و رفع بن بست

چشم پوشی:

چون احتمال بن بست پایین و هزینه رفع آن بالاست، می توان از آن چشم پوشی کرد و برخورد با آن را بر عهده برنامه نویس یا مدیر سیستم گذاشت.

فرصت :

تراکنش فقط برای مدت زمان معینی منتظر بماند و در صورت عدم اجابت، ساقط شود. این روش ساده است،اما تعیین این مقدار زمان مشکل است.

پیشگیری:

با این روش سیستم هرگز دچار بن بست نمی شود. پس:

1) می توان از پروتکل هایی مثل C2PL و SC2PL استفاده کرد، که هر تراکنش قبل از شروع تمام قفل های مورد نیازش را بگیرد.
2) می توان بین داده ها یک ترتیب در نظر گرفت و تراکنش ها فقط بر اساس این ترتیب مجاز به قفل کردن باشند.

اجتناب:

تراکنش ها اجرا می شوندو هنگام درخواست داده،احتمال وقوع بن بست بررسی می شود. در این حالت از مهر زمانی بن بست و یکی از روشهای زیر استفاده می شود:

- روش wait-die : تراکنش پیرتربا مهر زمانی کمترمنتظرتراکنش جوانتر می ماند تا قفل مربوطه را آزاد کند. در مقابل تراکنش جوانترهرگزمنتظر نمی ماند و ساقط می شود.(اولویت با تراکنش جوانتر)
- روش wound-wait : تراکنش پیرتر به جای انتظار،تراکنش جوانتر را می کشد و تراکنش جوانترمنتظرتراکنش پیرتر می ماند. (اولویت با تراکنش پیرتر)

تشخیص و رفع بن بست:

در این روش اگربن بست رخ دهد،مشخص وبرطرف می شود.

- برای تشخیص بن بست از گرافی به نام **گراف انتظار**استفاده می کنیم. این گراف،گره های آن تراکنش ها هستند و لبه Ti->Tjیعنی تراکنش Ti منتظر تراکنش Tj است. اگر حلقه در گراف وجود نداشته باشد، یعنی بن بست نداریم.

- در صورت وجود بن بست یا حلقه باید یک تراکنش در این حلقه را انتخاب وساقط کرد که به آن **قربانی** (victim) می گویند.

معیارهای انتخاب قربانی:

1) تعداد حلقه هایی که این تراکنش در آن وجود دارد و با ساقط شدن آن، این حلقه ها از بین می روند.
2) حجم کاری که تا کنون در آن تراکنش انجام شده است.
3) تعداد بروزرسانی انجام شده درآن
4) حجم کاری از تراکنش که برای اتمام آن باقی مانده است.