

روال دسترسی به داده ها برای انجام تراکنش:

عمده ترین عامل سرعت در پردازش اطلاعات، تعداد دفعات دستیابی به رسانه است.

برای کاهش تعداد دفعات دستیابی به رسانه، رکوردها را بلوک بندی می کنیم و در هر بار مراجعه به دیسک به جای یک رکورد یک بلوک خوانده می شود.

بافر: بخشی از حافظه به منظور پردازش داده های بانک اطلاعات

روال دسترسی به داده ها:

ناحیه کاری (work area): هر تراکنشی که می خواهد با داده ای کار کند، یک کپی از آن داده را در بافر در ناحیه کاری خاص خود می برد. در ناحیه کاری کلیه اطلاعات بروز و قابل دسترس توسط تراکنش نگهداری می شود.

انواع عملگرها:

Input(): انتقال داده ها از دیسک به بافر

output(): انتقال نتایج از بافر به دیسک

Read(): انتقال اطلاعات از بافر به ناحیه کاری تراکنش ها

Write(): انتقال اطلاعات از ناحیه کاری به بافر

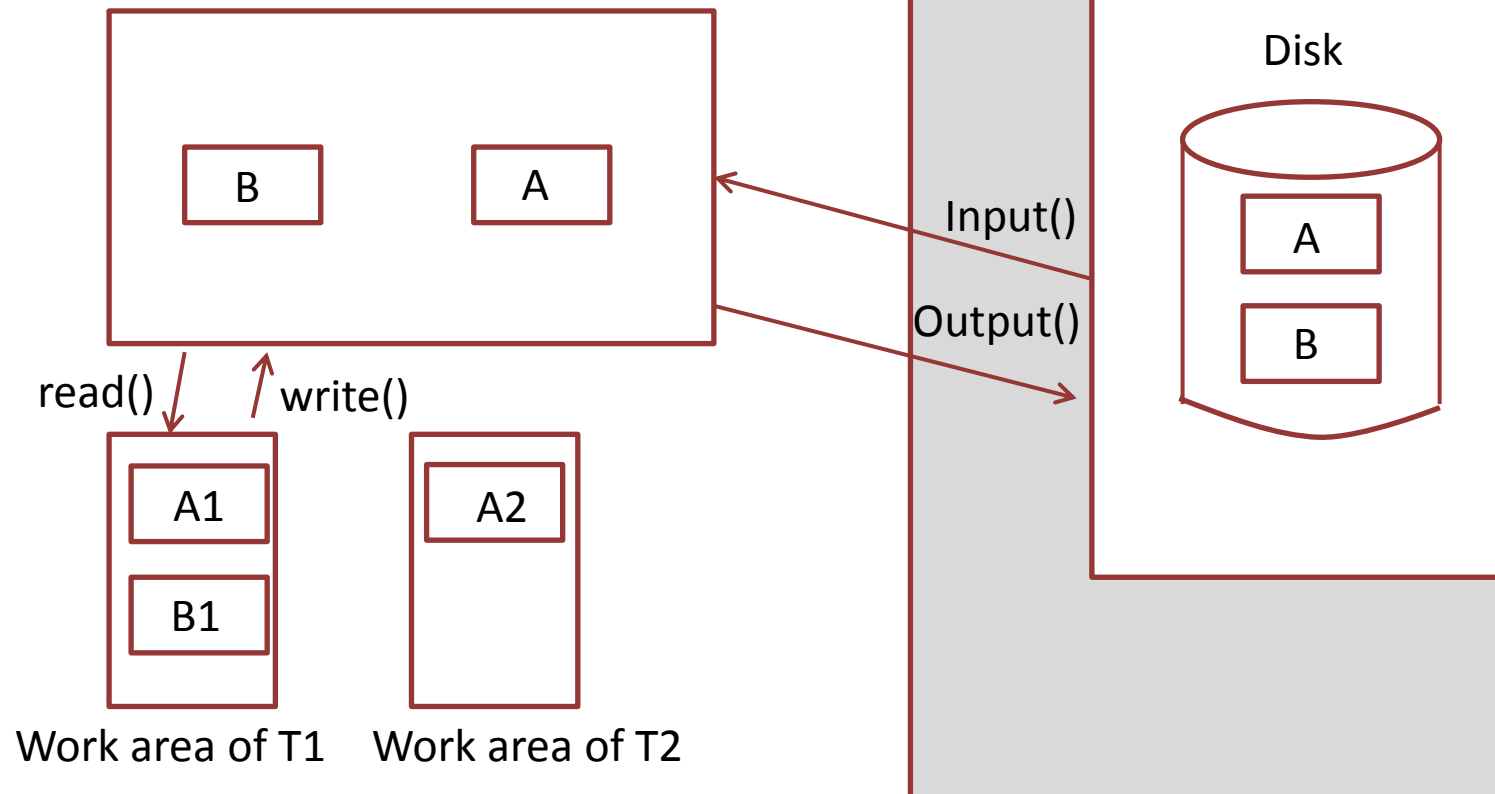
روال دسترسی به داده ها:

حافظه RAM

رسانه

Buffer

Disk



روال دسترسی به داده ها:

در هر سیستمی تعداد بافرها و سایز آنها متفاوت است.

بنابر این سایز بلوکهایی که در هر بافر نوشته می شود و یا به عبارتی تعداد رکوردهایی که در هر بلوک نوشته می شود متفاوت است.

فضای بافر را از دست داده ایم. $\text{Size}(\text{block}) < \text{size}(\text{buffer})$

باید بافرها را به هم چسباند و آنها را خواند. $\text{Size}(\text{block}) > \text{size}(\text{buffer})$

در نتیجه باید: $\text{size}(\text{block}) = \text{size}(\text{buffer})$

الگوریتم های ترمیم:

برای تضمین جامعیت بانک اطلاعاتی و اعمال خواص یکپارچگی و پایایی تراکنش ها در صورت وقوع خرابی از الگوریتم هایی استفاده می کنیم که شامل دو مرحله است:

1) در حین عملکرد عادی سیستم، اطلاعاتی که برای انجام عمل ترمیم (پس از وقوع خرابی) به آنها نیاز داریم در جایی ثبت می شود.

2) پس از وقوع خرابی (وقتی سیستم دوباره بالا می آید)، با استفاده از اطلاعات ثبت شده در مرحله قبل، خواص یکپارچگی و پایایی تراکنش ها اعمال گردند.

پس در مرحله از الگوریتم ترمیم، هر تراکنشی که انجام (commit) شده است را تکرار (redo) کرده و هر تراکنشی که ساقط شده را خنثی (undo) می کنیم. فرض می کنیم تراکنشها پی در پی هستند.

الگوریتم های ترمیم: (ادامه)

❖ دو روش کلی زیر برای ترمیم و بازیابی وجود دارد:

(1) رویکرد کارنامه (log-based)

(2) رویکرد رونوشت (shadow-paging)

رویکرد کارنامه:

در این رویکرد اطلاعات مورد نیاز برای انجام ترمیم را در حافظه پایدار ثبت می کنیم.

برای هر یک از دستورات یک تراکنش، رکوردی به نام رکورد کارنامه (log) با ساختار زیر نوشته می شود:

(1) برای شروع تراکنش T_i ، رکورد $\langle T_i, \text{start} \rangle$

(2) برای دستور $\text{write}(x)$ از تراکنش T_i ، در حالت کلی رکورد $\langle T_i, x, v_1, v_2 \rangle$ که v_1 و v_2 به ترتیب مقادیر قبل و بعد از انجام نوشتن x می باشند.

(3) با اجرای آخرین دستور تراکنش T_i ، رکورد $\langle T_i, \text{commit} \rangle$ در کارنامه ثبت می شود.

انعکاس تغییرات روی بانک اطلاعات:

❖ انعکاس تغییرات روی بانک اطلاعات به دو روش انجام می شود:

(1) انعکاس معوق تغییرات در بانک اطلاعات

(deferred database modification)

(2) انعکاس فوری تغییرات در بانک اطلاعات

(immediate database modification)

انعکاس معوق تغییرات روی بانک اطلاعات:

- ❖ طرح انعکاس معوق تغییرات در بانک اطلاعات، کلیه تغییرات را در رکوردهای کارنامه ثبت می نماید، اما کلیه عملیات نوشتن را به بعد از پذیرش ناقص تراکنش (Partial commit) موکول می نماید.
- ❖ فرض کنید تراکنش ها به صورت پی در پی انجام می شوند.
- ❖ تراکنش با عمل نوشتن رکورد کارنامه به شکل $\langle T_i, \text{start} \rangle$ ، شروع می شود.
- ❖ نتیجه عمل نوشتن آیتم داده ای x ($\text{write}(x)$)، ثبت یک رکورد کارنامه به شکل $\langle T_i, x, v \rangle$ خواهد بود، که در آن v مقدار جدید آیتم داده ای x می باشد.
- ❖ در حال حاضر، عمل نوشتن آیتم داده ای x ، قابل انجام نمی باشد، ولی به تاخیر افتاده است.
- ❖ هنگامیکه تراکنش T_i به طور ناقص به اتمام می رسد (partially committed) یک رکورد \log به شکل $\langle T_i, \text{commit} \rangle$ ثبت خواهد شد.
- ❖ در نهایت، رکوردهای \log برای اجرای واقعی عملیات نوشته شدن های به تاخیر افتاده قبلی، خوانده و مورد استفاده قرار می گیرند.

انعکاس معوق تغییرات روی بانک اطلاعات:

❖ در هنگام ترمیم و بازیابی بعد از یک توقف سیستم (crash)، یک تراکنش نیاز به اجرای مجدد (redo) دارد، اگر و فقط اگر هر دور کورد log شامل $\langle Ti, start \rangle$ و $\langle Ti, commit \rangle$ ثبت شده باشند.

❖ اجرای مجدد تراکنش T_i ، مقدار تمامی داده های منعکس شده توسط تراکنش T_i را به مقدار جدید، تنظیم می کند.

❖ خطای توقف در دو حالت زیر رخ می دهد.

(1) تراکنش در حال ویرایش اصلی مربوط به دستورات خودش باشد.

(2) فعالیت ترمیم در حال انجام باشد.

انعکاس معوق تغییرات روی بانک اطلاعات:

❖ اگر رکورد log در هنگام خطای توقف، روی حافظه ذخیره سازی پایدار وجود داشته باشد:

- 1) نیاز به هیچ اجرای مجددی نیست. (redo T0/redo T1)
- 2) از آنجاییکه رکورد کارنامه به شکل $\langle T0, commit \rangle$ ثبت شده است، بنابراین تراکنش T0 ضروری می باشد.
- 3) از آنجاییکه رکورد کارنامه به شکل $\langle T1, commit \rangle$ و $\langle T0, commit \rangle$ ثبت شده اند، بنابراین نیاز به اجرای مجدد ابتدا تراکنش T0 و سپس تراکنش T1 است.