MATHEUS PEIXOTO RIBEIRO VIEIRA - 22.1.4104

VITOR OLIVEIRA DINIZ - 22.1.4107

22.1) Para o protocolo de bloqueio em duas fases, deve ser realizada uma operação de read_lock(x) para realizar uma leitura e um write_lock(x) para realizar a escrita. Neste protocolo, ocorre uma fase de expansão, quando são realizados os bloqueios, e, depois, ocorre a contração, onde ocorrem os desbloqueios.

Como os itens são bloqueados antes de serem utilizados, qualquer outra transação que tente utilizar este mesmo item deverá esperar que ele seja desbloqueado. Então, qualquer escalonamento que for feito, deverá levar em conta a espera do bloqueio. Assim, como o desbloqueio só é feito após o fim do uso do recurso, então, não ocorre o conflto. Por fim, pode ser que ocorra conflitos da transação 1 na transação 2, mas não irá ocorrer o conflito da transação 2 na transação 1, evitando ciclos no grafo de precedencia.

22.5) Em ambos os protocolos, levamos em consideração o timestamp das transações em questão. No protocolo esperar-morrer, caso a transação mais recente tente acessar um item bloqueado, ela será abortada, e caso a transação mais antiga tenta acessar o item, para ela será permitido a espera.

No protocolo ferir-esperar, caso a transação mais velha tenta acessar o item bloqueado pela mais nova, a mais nova será abortada. Agora, caso a transação mais nova tente acessar o item, ela será liberada para esperar.

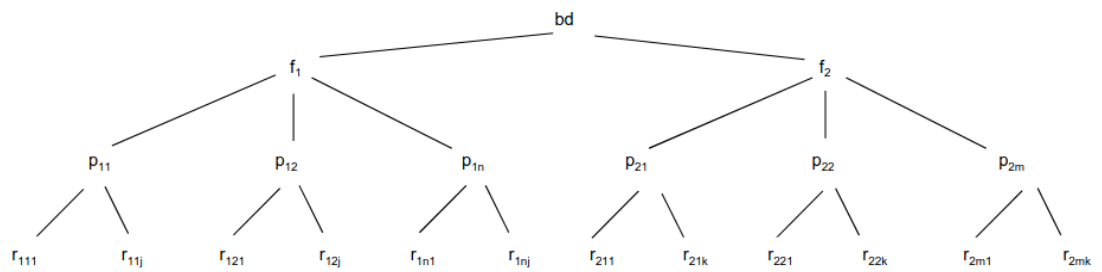
22.7) O timestamp é um identificador único gerado pelo SGBD para identificar uma transação, sendo gerados por um contador incremental ou utilizam o valor corrente do relógio do sistema, variando em data, hora, segundo, milissegundo.

22.15) Suponha uma árvore de granularidade múltipla e que uma modificação está sendo feita em um registro da árvore, então ele está bloqueado para escrita. Agora,

vamos supor que um nó pai deste registro, um arquivo, vai ser modificado, para isso, deveríamos percorrer todos os filhos deste arquivo para, assim, perceber que há um bloqueio de escrita ativo. Dessa forma, como solução, foi criado o bloqueio de inteção. Assim, saindo do nó raiz e indo até o nó desejado, vai sendo marcada a intenção de bloqueio, sendo ela de escrita, leitura ou uma leitura que se tornará escrita. Portanto, quando a situação descrita acima ocorrer, iremos verificar somente até o registro mais perto da raiz, evitando muitas verificações.

22.27) No bloqueio em duas fases, todos os bloqueios são realizados antes do primeiro desbloqueio. Suponha agora que um item deseja ser acessado para escrita. Dessa forma, o nó raiz seria bloqueado em modo exclusivo, em seguida o nó filho e assim por diante até encontrar o dado desejado ou chegar em um nó nulo, uma vez que, caso ele esteja completo, modificações serão feitas nos nós superiores. Assim, após encontrar o nó adequado e inserir as informações que começaremos a desbloquear os nós, fazendo com que outras transações esperem por este desbloqueio, de modo que nenhuma operação possa ser realizada, o que se torna um processo muito lento, que não ocorre na implementação devida, já que, ao perceber que o nó pai não será modificado, ele já é desbloqueado.

22.28) Saindo do nó raiz e indo até os nós desejados, marcando os nós no caminho com a intenção de escrita/leitura, podemos ter um caminho onde ambas as inteções irão existir, porém, elas irão divergir em algum momento, onde uma irá para um item que realizará a escrita, enquanto a outra irá para um item que realizará a leitura. Dessa forma, somente compartilham um mesmo caminho, mas tendo as operações realizadas em recursos diferentes, não havendo concorrência de dados.



Supondo um exemplo com a árvore de hierarquia de granularidade do slide 51 da aula de controle de concorrência, pode ser que o registro r_{111} será escrito e o registro r_{11j} será lido. Dessa forma, os nós bd, f_1 e p_{11} devem possuir tanto o bloqueio IS quanto o IX, pois indicam que os filhos irão executar alguma dessas ações.