1.3 Producer-Consumer проблема

Взаємне виключення не єдина проблема що заслуговує вивчення. Зрештою, Аліса і Боб закохуються і одружуються. Зрештою, вони розлучаються. (Про що вони думали?) Суддя дає дозвіл Алісі на зберігання домашніх тварин, і каже Бобу, щоб нагодував їх. Домашні тварини в даний час ладнають один з одним, але вони опліч з Алісою, і атакують Боба щоразу, коли вони бачать його. В результаті, Аліса та Боб повинні розробити протокол для Боба щоб доставити їжу до домашніх тварин без участі Боба і щоб домашні улюбленці були разом у дворі. Крім того, даний протокол не повинен витрачати чий-то час: Аліса не повинна випускати своїх домашніх тварин до двору без їжі й Боб не повинен входити до двору якщо тварини не спожили всі продукти харчування. Ця проблема відома як Producer-Consumer проблема.

Дивно, консервні банки що ми не визнали для взаємного виключення не являється саме тим що нам потрібно для Producer-Consumer проблеми. Боб ставить банку на підвіконня Аліси, прив’язує один кінець навколо банки й кладе інший кінець в своїй вітальні. Потім він кладе їжу у дворі й вибиває банку. Тепер, коли Аліса хоче випустити своїх домашніх тварин вона робить наступне:

1. Вона чекає допоки банка впаде.
2. Вона випускає улюбленців.
3. Коли улюбленці повернулись, Аліса перевіряє чи все вони з‘їли. Якщо так, вона заново ставить банку.

Боб робить наступне:

1. Він чикає допоки банку піднімуть.
2. Залишає їжу у дворі.
3. Він тягне стрічку й вибиває банку.

Стан банки відображає стан двору. Якщо банка впала, це означає що є їжа і тваринки можуть обідати, і якщо банка стоїть, це означає що їжа закінчилася й Боб може принести ще. Перевіримо наступні три властивості:

* Взаємне виключення: Боб й тварини не можуть бути у дворі разом.
* Starvation-freedom: Якщо Боб завжди готовий годувати, й тварини зазвичай голодні, то домашні улюбленці можуть нескінченно часто їсти.
* Producer-Consumer: Домашні тварини не будуть входити в двір якщо немає їжі, Боб ніколи не додасть їжі якщо ще залишилася не з’їдена їжа.

Producer-Consumer протокол й протокол взаємного виключення розглянуті в попередній секції запевняють що Боб та Аліса ніколи не будуть разом у дворі в один й той самий час. Проте, Аліса та Боб не можуть використовувати Producer-Consumer протокол для взаємного виключення, важливо розуміти чому. Взаємне виключення вимагає глухий кут: кожен може увійти в двір нескінченно часто, якщо іншого тут немає. На відміну, Producer-Consumer протокол передбачає безперервну співпрацю з обох сторін.

Ось як ми розмірковуємо на рахунок даного протоколу:

* Взаємне виключення: Ми використовуємо дещо інший стиль для доведення того що ми використовуємо для взаємного виключення: «машина станів» на основі доказів, на відміну від суперечності. Ми розмірковуємо на стрічці від банки як о машині станів. Вона може мати два стани, вгорі та внизу, і неодноразово переходить між цими станами. Ми вважаємо, що взаємне виключення має місце, так як він тримає спочатку і продовжує утримувати при переході з будь-якого стану банки в інші.

Спочатку банка або вгорі або внизу. Скажемо що вона внизу. Тому одна з домашніх тварин може зайти, і взаємне виключення підтверджується. У випадку коли банка може бути піднята Алісою, тварини повинні спочатку вийти, й коли банка піднята, тварини не знаходяться у дворі й взаємне виключення зберігається до того моменту поки вони знову не увійдуть. У випадку коли банка вибита, Боб повинен покинути двір й не повинен входити в нього до того моменту поки банка знову не буде піднята, тому взаємне виключення зберігається після того як банка вибита. Немає інших можливих станів, тому наше твердження вірне.

* Starvation-freedom: допустимо наше твердження не вірне: це повинен бути випадок коли нескінченно часто домашні улюбленці Аліси голодні, немає їжі, й Боб намагається надати їжі, але ці намагання не успішні. Банка не може бути піднята, так як Боб намагається дати їжі і вибиває банку, дозволяючи тваринам поїсти. Тому банка повинна бути внизу, й коли домашні улюбленці голодні, Аліса піднімає банку, повертаючи все до першого випадку.
* Producer-Consumer: взаємне виключення властивостей означає що домашні улюбленці й Боб ніколи не будуть у дворі разом. Боб не зайде у двір допоки Аліса не підніме банку, що вона робить тільки якщо більше не вистачає їжі. Аналогічно, тваринки не входять у двір допоки Боб кладе банку, що він робить тільки після того як доставить їжу.

Як і протокол взаємного виключення що ми описали раніше, цей протокол показує затримки. Якщо Боб поклав їжу у дворі, й негайно пішов не піднявши банку, тварини можуть зголодніти, не зважаючи на те що у дворі є їжа.

Повертаючись до комп’ютерної науки, Producer-Consumer проблема з’являється майже у всіх паралельних й розподілених системах. Цим шляхом процесори надають інформацію в буфери обміну для читання або переміщення через мережеві з’єднання або спільні шини.

1.4 Readers–Writers проблема

Боб та Аліса несподівано вирішили що люблять своїх домашніх улюбленців так сильно що їм необхідно обмінюватись повідомленнями про свої тварин. Боб ставить рекламний щит перед своїм домом. Рекламний щит тримає великі плитки у певному порядку, кожна з яких містить єдину літеру. Боб, у свій вільний час, виставляє повідомлення на рекламному щиті шляхом піднімання однієї плитки в одиницю часу. Аліса, в свій вільний час, читає повідомлення дивлячись на рекламний щит через телескоп, кожну плитку в одиницю часу.

Це може виглядати як працююча система, але це не так. Уявіть що Боб виставляє повідомлення:

«Продам кицю»

Аліса, дивлячись через телескоп, переводить повідомлення:

«Продам»

На цьому моменті Боб забирає плитки й пише нове повідомлення:

«Помию собаку»

Аліса, продовжуючи сканувати через рекламний щит дочитує повідомлення:

«Продам собаку»

Ви можете уявити й інше.

Є декілька простих шляхів вирішити дану проблему.

* Аліса і Боб можуть використовувати протокол взаємного виключення задля того щоб бути впевненим що Аліса читає тільки закінченні речення. Вона може пропустити повідомлення, однак.
* Вони можуть використовувати проток «консервних банок», де Боб пише речення й Аліса читає їх.

Якщо дану проблему так легко вирішити, чому ми розглядаємо її? Обидва протоколи вимагають затримки: якщо один учасник піддається несподіваній затримці, то інший також. В контексті загальної багатопроцесорної пам’яті, вирішенням даної проблеми є можливість потоків захопити частину декількох осередків пам’яті. Захопивши таку частину, не чекаючи, тобто, не заважаючи іншим потокам модифікувати ці місця під час їх читання, є потужним інструментом, який може бути використаний для резервного копіювання, налагодження і в багатьох інших ситуаціях. Дивно, але дана проблема має вирішення які дозволяють обійтись без затримок. Оглянемо кілька таких рішень пізніше.