

**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

НАПРАВЛЕНИЕ „ КОМПЮТЪРНО И СОФТУЕРНО ИНЖЕНЕРСТВО”

КУРСОВА РАБОТА

ПО ПС

ИЗРАБОТИЛ: БОРИСЛАВ ФИЛИПОВ ФИЛИПОВ

Група: 43

Факултетен номер: 121213101

София 2016

1. **Увод**

Условие:

7. Да се реализира приложение, което "изтегля" торент по зададен адрес. Задайте директория, където искате този торент да бъде изтеглен и запишете във файл името на изтегленото съдържание (напр. "GTA V"), неговия размер и адреса, от който е дошъл. Реализирайте две програми, като първата е клиент (съдържа името на потребителя и адреса на файла, реализира записа на торента на PC). Тя изпраща тези данни до сървъра (друга програма), а той от своя страна записва във файл всички изтегляния, които потребителите са направили, и при ново стартиране на програмата, ги възстановява в свой контейнер. Ако потребител иска да направи отново дадена заявка, то трябва да се изпише, че тя вече е извършена и не може да бъде направена отново. Демонстрирайте работа с повече от един клиенти.

1. Клиента се свързва със сървара като му изпраща име и уеб адрес на файла.
2. Сървъра комуникира с едновременно с много клиенти;
3. Сървъра проверява дали клиент с това име е свалял вече този файл и връща отговор към клиента.
4. Ако файлът не е бил свалян от клиент с такова име, клиента сваля файла и го записва на диска.
5. Клиента записва в отделен файл данните за сваленото съдържание.
6. **Глава 1**
7. Клиента се свързва със сървър посредством сокет. Тази технология е най подходяща, защото е възможно сървъра да е на отдалечена машина.
8. Сървъра създава нишка за всеки клиент, която изпълнява функциите на сървъра. Нишките са най-подходящи, защото сървъра трябва да може да си комуникира с няколко клиента едновременно. Освен това всички нишки имат общ стек, което улеснява достъпа до общи ресурси.
   1. За всеки клиент се пази файл с адресите на всички файлове, които той е свалял.
   2. В началото се зарежда масив с всички юзъри. Към всеки юзър се закача отделен масив с адресите на файловете. Масивите са сортирани съответно по име на юзър и име на файл, за да може да се използва бързото бинарно търсене. Избраната структура е масив, защото е най-удачен при търсене. Въпреки, че добавянето на елементи е по-бавно в сравнение със други структури, общото време е по-малко.
   3. След получаване на заявка, чрез бинарно търсене се проверява дали дадения файл е свалян.
      1. Ако е свалян, се връща отговор към клиента.
      2. Ако не е свалян, се добавя в масива и се връща отговор към клинета.
      3. Ако потребителското име не съществува се добавя в масива с потребители и се връща отговор на клиента
   4. Масивите се синхронизират с файловата система.
9. При получен отговор следва:
   1. Ако файлът не е бил свалян, клиента отваря нов сокет към съответния HTTP адрес.
      1. Праща се GET заявка(спред HTTP протокола);
      2. Записва се файла на диска.
   2. Ако файлът е сваля, клиента връща съобщение към потребителя и спира своята работа.
10. Записват се данните за файла в общ файл на клиента.
11. **Глава 2**
12. Клиент
    1. Handle\_link()

Изкарва нужната информация от линка – адрес на сървъра, порт, път до файла и име на файла.

* 1. Download\_file()

Отваря сокет към уев сървъра. Комуникира си с него посредство HTTP. Записва файла на диска.

* 1. Write\_to\_file()

Записва данните засваления файл на нов ред в общия такъв.

* 1. Check\_with\_server()

Отваря сокет към сървъра. Изпраща потребителското си име и адрес на файла. Изчаква отговор от сървъра и връща резултат.

* 1. Main

Извиква функциите в следната последователност:

Check\_with\_server()

Ако резултата е положителен

Download\_file()

Write\_to\_file()

1. Сървър
   1. Users\_free
      1. Освобождава паметта ма масивите.
   2. User\_check
      1. Проверява дали дадения потребител е в масива посредством бинарно търсене
   3. User\_insert
      1. Добавя потребител в масива. Поддържа масива сортиран
   4. Files\_check
      1. Проверява дали дадения файл е в масива посредством бинарно търсене
   5. Files\_insert
      1. Добавя файл в масива. Поддържа масива сортиран
   6. Handle\_connection
      1. Поведението на всяка нишка от сървъра.
      2. Получава информация от клиента.
      3. Вика user\_check
         1. Ако не съществува
            1. User\_insert
         2. Ако съществува
            1. Вика files\_check

Ако не съществува

Files\_insert

Връща отговор, че файлът може да бъде свален

Ако съществува

Връща отговор, че файлът не може да бъде свален

* 1. Start\_server
     1. Приема конекции в цикъл
     2. За всяка конекция стартира нишка с handle\_connecton
  2. Main
     1. Извиква start\_server със съответните параметри.

Достъпа до масивите от нишките се синхронизира по следния начин:

* Следи се колко нишки обхождат масива на потребителите едновременно посредством брояч и mutex- за този брояч
* Когато трябва да се добави потребител:
  + съответната нишка добавя потребител в отделен свързан списък.
  + Блокира достъпа на всички останали нишки до масива посредством Condition variable.
  + Изчаква броят на нишките, които обхождат масива, да стане 0. Сигнализацията става посредством друг Condition variable
  + Ако се окаже, че някоя от нишките, които са вече вътре трябва да добавят нов потребител, те го добавят в списъка с потребители за добавяне, пращат отговор и приключват своята работа.
  + Когато броят на нишките, обхождащи масива, стане 0, първата нишка добавя всеки елемент от списъка в масива(user\_insert).
  + Посредством трети Condition variable се сигнализира, че масива е свободен и останалите нишки започват пак да го обхождат.
* При добавянето на файл, синхронизацията е аналогична за всеки от масивите с адреси на файлове.

1. **Глава 3**
   * + 1. Сървър:

Сървърът се стартира с 2 параметъра.

* + - * 1. Първият оказва порта на който ще слуша сървъра. Той трябва да е цяло неотрицателно число.
        2. Втория не е задължителен. Той показва колко конекции максимално може да приеме сървъра преди да се прекрати. Той трябва да е цяло неотрицателно число. Ако не се подаде нищо като втори параметър или се подаде 0, сървърът ще работи в безкраен цикъл.
      1. Клиент:

Клиента се стартира с 4 параметъра.

* + - * 1. Име или IP на сървъра
        2. Порт на сървъра
        3. Потребителско име
        4. Уеб адрес, от който ще се сваля файла

1. **Заключение**

Това е едно добро решение на проблема. Не са направени достатъчно тестова, за дасе установи дали това е най-доброто решение. Имплементиране е комуникация само с HTTP протокол.