

TabManager

TabManager представлява конзолно приложение, което симулира уеб браузър при работа с табове. Всеки таб съдържа информация за виртуалните интернет страници, съответно URL адрес и съдържание. Виртуалните страници са представени с помощта на текстови файлове, чието име е URL адреса, последван от .txt разширение, а текстовото съдържание на файловете е съдържанието на страниците. Всички файлове с виртуални страници се намират в директорията content. TabManager реализира всяка от следните функционалности:

Реализира се програма с команден интерфейс, в която могат да се въвеждат описаните по-долу команди. За всяка от тях е посочена сложността, с която тя да работи. В описанията на сложностите приемаме, че N е броят на текущо отворените табове.

GO url

Командата зарежда страницата с URL в текущия таб. Автоматично трябва да се обнови timestamp-а на таба. Сложност: $O(U+C)$, където U е дължината на , а C – тази на съдържанието му.

INSERT url

Има два варианта. (1) Ако URL-ът вече е зареден в някой таб на браузъра, този таб става текущ. Ако има няколко такива таба, текущ да стане първият от тях. (2) В противен случай, след текущия таб се добавя нов, в който се зарежда страницата с URL . За този нов таб автоматично се обновява неговия timestamp. Новият таб става текущ. Сложност: $O(U+C)$, където U е дължината на , а C – тази на съдържанието му.

BACK Потребителят се прехвърля на таба, който е непосредствено преди текущия. Ако такъв няма (в момента сме върху първия таб), не се случва нищо. Сложност: $O(1)$

FORWARD Браузърът отива на таба, който е непосредствено след текущия. Ако такъв няма (в момента сме върху последния таб), не се случва нищо. Сложност: $O(1)$

REMOVE Премахва текущия таб. Текущ става този след него. Ако такъв няма, текущ става този пред него. Ако това е бил последният таб, отваря се нов и в него се зарежда адрес "about:blank". Сложност: $O(1)$

PRINT Извежда на екрана информация за всички табове. Форматът е следния: (Между двата елемента има един интервал) Редът, на който се намира текущия таб, трябва да започва със знак за по-голямо ($>$), например: somesite.com

123123123 > www.example.com 123456789 Somethingelse.com/index.html
123123321

Табовете да се извеждат точно в реда, в който се пазят в брауъра. Сложност: $O(N)$

SORT by

Сортира всички табове лексикографски и прави текущ първия в наредбата. може да приема две стойности: * URL – сортира табовете по URL, като ако има два с еднкъв URL, те се подреждат по timestamp; * TIME – сортира табовете по техния timestamp, като ако има два с еднакъв timestamp, те се подреждат по URL. Сложност: Да се използва алгоритъм за сортиране със сложност $O(N \log N)$. Крайната сложност на сортирането може да бъде по-голяма, тъй като сравнението на низове не винаги е бърза операция. Крайният резултат трябва да бъде със сложност не по-лоша от $O(UN \log N)$, където U е максималната дължина на URL от тези заредени в брауъра.

SEARCH word

По подадена дума , търси във всички отворени табове и извежда на екрана списък на онези от тях, които съдържат в своето съдържание. Табовете в резултата трябва да се изведат номерирани от 1 до N . След това потребителят трябва да може да избере един от тях и той става текущ. Сложност: Не повече от $O(W + T \log(T))$, където W е дължината на , а T е броят на табовете, които я съдържат.

Зареждането на съдържанието на страници трябва да става по следния начин: приемаме, че в работната директория на програмата (обикновено същата, от която я стартираме), ще има поддиректория content, която съдържа текстови файлове. Когато зареждаме URL , програмата ви трябва да опита да зареди файл с име „content/.txt” и приемаме, че неговото съдържание е съдържанието на страницата. Например ако потребителят реши да зареди “example.com” и след това “ABC”, програмата ви трябва да опита да зареди съдържанието на следните файлове:

content/example.com.txt

content/ABC.txt

Ако при зареждане на даден URL, съответният му файл не бъде намерен или възникне проблем при прочитането му, приемаме, че страницата е празна и съдържанието ѝ ще бъде празният низ.

Реализация на приложението чрез следната архитектура:

