

Konu : Cache Buffer (Önbellek Tamponu) Tasarımı

Problem: Bu ödevde, double linked list (ikili linkli liste) veri yapısını kullanarak bir cache buffer sistemi tasarlamanız ve gerçeklemeniz istenmektedir. Gerçekleştireceğiniz cache buffer, bir network ortamında yapılan web sayfası isteklerini, hızlı ve kaliteli hizmet verebilmek için depolamak ve servis için önceliklendirmek amacı ile kullanılacaktır.

Ödev 4 ana bölümden oluşmaktadır:

1. **Double linked list** düğümlerinin ve veri yapısının gerçekleştirilmesi:
Her bir düğümde; istenen sayfanın adresi, toplam istek sayısı (sayacı), önceki ve sonraki düğümlerin işaretçileri tutulmalıdır. Düğümler üzerinde gezinti, sayaç sorgulama, düğüm ekleme ve silme gibi işlemler gerçekleştirilmelidir.
2. Verilen işlemin **double linked list** kullanılarak çözümü:
 - a. Giriş bilgisinin okunması:
 - i. İstek yapılan sayfaları string olarak okuyunuz (Ör: ABC, ABE, BCD, ...). Sayfalar bir metin belgesinden satır satır da okunabilmelidir.
 - b. Girdinin işlenmesi:
 - i. Girilen sayfa (veya dosyadan okunan 1 satır) önceden cache buffer'da yoksa yeni bir düğüm olarak yaratılır, sayacı 1 yapılır ve başlangıç düğümü (head node) olarak cache buffer'a eklenir.
 - ii. Girilen sayfa cache buffer'da mevcut ise, ilgili düğümün sayacı 1 arttırılır. Sayaç belirlenen bir **eşik değerinin** (Ör: $T = 10$) üzerine geçerse, bu düğüm cache buffer'da **head node** olarak atanır. (Önceki ve sonraki düğümlerin işaretçileri de düzenlenir.)
 - c. Cache buffer'ın bakımı:
 - i. Cache buffer'ın **belirli bir kapasitesi** vardır (Ör: $L=20$). Bu kapasite aşılsa, cache buffer'ın **son elemanı (düğümü) silinir** (2.b maddesinde kontrol edilmelidir).
3. Cache buffer'ın hizmet vermesi: Her yeni istek geldiğinde, 2.b maddesi işletilir ve cache buffer'ın mevcut hali linkli liste şeklinde, düğümlerin içerikleri ile (adres ve sayacı) ekranda gösterilir.
4. Cache Buffer'ın Temizlenmesi: Kullanıcıya listeleme işleminden sonra cache'i temizlemek isteyip istemediği sorulmalıdır. Temizlemeye onay verilirse listedeki tüm elemanlar silinmelidir.

Örnek:

(Yer tutmaması için girdiler satırlarda değil, yanyana ve birer karakter olarak gösterilmiştir.)

$T=5, L=4$

İSTEKLER: A B B A C C A A A C D E F D G D B

Ekran çıktıları: (parantez içindeki kırmızı ifadeler açıklama için eklenmiştir)

A,1 (A geldi, yeni düğüm head node atandı)

B,1 \rightarrow A,1 (B geldi, yeni düğüm head node atandı)

B,2 \rightarrow A,1 (B geldi, sayacı arttı)

B,2 \rightarrow A,2 (A geldi, sayacı arttı)

C,1 \hookrightarrow B,2 \hookrightarrow A,2 (C geldi, yeni düğüm head node atandı)
C,2 \hookrightarrow B,2 \hookrightarrow A,2 (C geldi, sayacı arttı)
C,2 \hookrightarrow B,2 \hookrightarrow A,3 (A geldi, sayacı arttı)
C,2 \hookrightarrow B,2 \hookrightarrow A,4 (A geldi, sayacı arttı)
C,2 \hookrightarrow B,2 \hookrightarrow A,5 (A geldi, sayacı arttı)
A,6 \hookrightarrow C,2 \hookrightarrow B,2 (A geldi, sayacı arttı, [T=5] A,6 head node olarak atandı)
A,6 \hookrightarrow C,3 \hookrightarrow B,2 (C geldi, sayacı arttı)
D,1 \hookrightarrow A,6 \hookrightarrow C,2 \hookrightarrow B,2 (D geldi, yeni düğüm head node atandı)
E,1 \hookrightarrow D,1 \hookrightarrow A,6 \hookrightarrow C,2 (E geldi, yeni düğüm head node atandı [L=4] son (5.) düğüm silindi)
F,1 \hookrightarrow E,1 \hookrightarrow D,1 \hookrightarrow A,6 (F geldi, yeni düğüm head node atandı [L=4] son (5.) düğüm silindi)
F,1 \hookrightarrow E,1 \hookrightarrow D,2 \hookrightarrow A,6 (D geldi, sayacı arttı)
G,1 \hookrightarrow F,1 \hookrightarrow E,1 \hookrightarrow D,1 (G geldi, yeni düğüm head node atandı [L=4] son (5.) düğüm silindi)
G,1 \hookrightarrow F,1 \hookrightarrow E,1 \hookrightarrow D,2 (D geldi, sayacı arttı)
B,1 \hookrightarrow G,1 \hookrightarrow F,1 \hookrightarrow E,1 (B geldi, yeni düğüm head node atandı [L=4] son (5.) düğüm silindi)

Ekrana Çıktıları: Girişte verilen ifadeyi string olarak adım adım veya metin dosyasından bir seferde alınız. Yapılan her istek sonunda cache buffer'ın mevcut linkli liste yapısını, düğümlerin içerikleri ile ekrana yazdırınız.

Teslim İşlemleri :

1. Ödevler 04 Mart 2020 Çarşamba günü yapılacak laboratuvarında değerlendirilecektir.
2. Raporunuzda:
 - a. Çözümünüze yönelik raporlama kurallarına uygun olarak hazırlanmış açıklama/diyagram/şekil/tablolara
 - b. Ödevinize ait C dilinde yazılmış program kodu
 - c. Aşağıdaki örnek(ler) için programınızın ekran çıktısı olmalıdır.

GİRİDİ(Input.txt)	ÇIKTI
T=2 , L=3 AB BA CY AB CY XYZ BA XYZ BA	AB,1 BA,1 \hookrightarrow AB,1 CY,1 \hookrightarrow BA,1 \hookrightarrow AB,1 CY,1 \hookrightarrow BA,1 \hookrightarrow AB,2 CY,2 \hookrightarrow BA,1 \hookrightarrow AB,2 XYZ,1 \hookrightarrow CY,2 \hookrightarrow BA,1 XYZ,1 \hookrightarrow CY,2 \hookrightarrow BA,2 XYZ,2 \hookrightarrow CY,2 \hookrightarrow BA,2 BA,3 \hookrightarrow XYZ,2 \hookrightarrow CY,2
T=3, L=4 A B A AA BBB B A AB A B A BB	?

3. 03 Mart 2020 Salı günü saat 23.59'a kadar aşağıdaki adrese yükleyiniz:

<https://forms.gle/L2EWWYMTtAApetdD6>

4. Ödev notunun %80'i algoritma %20'si rapor dokümantasyonu üzerinden verilecektir.