



GAZİ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ

BM311
BİLGİSAYAR MİMARİSİ

VIRTUAL MEMORY KULLANILDIĞINDA UYGULAMALAR
ARASINDA GEÇİŞTE VIRTUAL CACHE İLE KARŞILAŞILAN
PROBLEMLER VE KULLANILAN ÇÖZÜMLER

Hazırlayan
Dilay Ece MARAL
21118080093

28.10.2024

İÇİNDEKİLER

İÇİNDEKİLER	i
ARAŞTIRMA.....	1
1.Araştırma Konusu.....	1
2.Virtual Memory.....	1
3.Virtual Cache.....	1
4.Problemler ve Çözümler.....	2
4.1 Güvenlik(protection).....	3
4.1.1 Problem.....	3
4.1.2 Çözüm.....	3
4.2 Alias(synonyms)	4
4.2.1 Problem.....	4
4.2.2 Çözüm.....	4
4.3 Cache flushing.....	4
4.3.1 Problem.....	4
4.3.2 Çözüm.....	5
4.4 Homonym.....	5
4.4.1 Problem.....	5
4.4.2 Çözüm.....	5
DEĞERLENDİRME.....	6
KAYNAKÇA.....	7

ARAŞTIRMA

1.Araştırma Konusu

Araştırmanın konusu, virtual memory(sanal bellek) kullanıldığında uygulamalar arasında geçiş (context switch) sırasında virtual cache ile karşılaşılan sorunlar ve bu sorunların çözüm yöntemleridir. Uygulamalar arasında geçiş yapıldığında, sanal adreslerin fiziksel adreslere çevrilmesi ve cache yapılarına erişimi sırasında güvenlik, performans kaybı ve alias gibi problemler ortaya çıkabilir. Bu araştırma, özellikle alias sorunu, TLB geçersizliği, homonyms sorunu ve veri tutarsızlığı problemi gibi başlıca sorunları ele alacak ve bu sorunlara yönelik çözümleri inceleyecektir. Amaç, bu problemlerin nasıl yönetildiğini ve optimize edildiğini açıklamaktır.

2.Virtual Memory

Virtual memory, bilgisayar sistemlerinde fiziksel belleğin yetersiz kaldığı durumlarda belleğin verimli kullanılması için kullanılan bir tekniktir. Bilgisayarın RAM kapasitesini artırmak yerine, depolama alanını (genellikle sabit disk veya SSD) geçici bir RAM alanı olarak kullanarak programların kesintisiz çalışmasını sağlar. Virtual memory, fiziksel belleğin bir kısmını diskte bir alan olarak ayırır ve ihtiyaç duyulduğunda bu alanı RAM olarak kullanır. Bu işlem, bilgisayarın aynı anda birden çok programı çalıştırmasına ve büyük veri setlerini işleyebilmesine olanak tanır (Silberschatz et al., 2018, pp. 389–390).

3.Virtual Cache

Önbellek, RAM'deki verilere doğrudan erişmek yerine sık kullanılan verileri tutarak CPU'nun işlem yaparken daha hızlı veri erişimi sağlamasına olanak tanıyan küçük ama hızlı bir bellek türüdür. Önbellekte belleğe erişebilmek için işlemciden gelen sanal adreslerin fiziksel adreslere dönüştürülmesi gerekir. Bu dönüşümün işleyişi bakımından iki ayrı önbellek türü ortaya çıkar: physical cache ve virtual cache.

Virtual cache ile physical cache'in farkı, virtual cache'in işlemci belleğe erişmek istediğinde önce sanal adresin (virtual address) önbellekte olup olmadığına bakmasıdır; eğer

önbellekte eşleşen bir satır bulursa fiziksel adresi (physical address) tekrar çevirmez. Her seferinde bir dönüştürme işlemi zaman alacağından, sık kullanılan durumların daha hızlı yapılabilmesi için önbellekte sanal adreslerin kullanılması daha verimli olur. Bunun ana sebebi, hit (verinin bulunma) olasılığının miss (verilerin bulunamaması) olasılığından çok daha yüksek olmasıdır.

Önbellek yönetiminde ise iki önemli görev karşımıza çıkar: indeksleme(indexing) ve comparing address. İndeksleme kısaca önbellek içinde verilerin nasıl organize edildiğini belirler, adres karşılaştırma ise önbellek içinde bir veri parçasının geçerli olup olmadığını kontrol etmek için yapılan bir işlemdir. Bu işlemlerde sanal veya fiziksel adres kullanılıp kullanılmaması ise önemlidir. Eğer bu işlemlerin ikisinde virtual address kullanılırsa, Full Virtual Addressing ile adres çevirme süresini ortadan kaldıracabiliriz fakat bu yöntemin her zaman tercih edilmemesinin bazı sebepleri var (Hennessy & Patterson, 1989, p. B-36-B-38).

4.Problemler ve Çözümler

Çok sık tercih edilmemesine sebep olan problemler şunlardır: güvenlik problemi, alias (eşanlımlılar) olarak geçen birden fazla aynı sanal adres (virtual address) olma durumu, I/O cihazlarının yalnızca fiziksel adres (physical address) kullanması, homonym olarak bilinen tek bir sanal adresin iki veya daha fazla farklı fiziksel adrese eşlenmesi durumu ve context switch (bağlam değiştirme) sırasında yaşanan bir takım sorunlar. Bu araştırmanın konusu, context switch sırasında oluşan problemler olup, bu işlem sırasında yukarıda bahsedilen sorunlardan bazıları da görülebilmektedir.

Kısaca, context switch olarak bilinen durum, yapılan işin değişmesidir. Bir işlemten diğerine geçiş yapılırken, sanal adreslerin (virtual addresses) her işlem için farklı fiziksel adreslere (physical addresses) referans vermesi gerekir. Bahsedilecek olan problemlerin bazıları birbirini kapsayabilecek olup aynı zamanda başka problemlerin çözümü olarak da kullanılabilir. Buna örnek olarak ele alacağımız problemlerden olan her işlem değişiminde önbelleğin (cache) temizlenmesi (cache flushing) ve ardından yeni verilerin alınması (cache refill) gerekliliği verilebilir. Bir diğer problem olan güvenlik problemi context switch için de geçerlidir çünkü sayfa düzeyinde koruma ve koruma bilgileri fiziksel adresin çevrilmesi sırasında kontrol edilir fakat virtual cache doğrudan sanal adres kullanır. Bunların yanında alias veya synonyms (eşanlımlılar) olarak geçen aynı sanal adresin (virtual address) birden fazla olması durumu da önemli bir problemdir. Context switch sırasında bir işlemten diğerine geçerken, yeni işlemle aynı sanal adresleri kullanan ancak farklı fiziksel adreslere işaret eden eski önbellek (cache) içeriği bulunabilir. Bu, virtual cache'in doğru veriyi tutamamasına sebep olabilir. Başka bir problem ise

homonym problemidir ve benzer isimlerin veya terimlerin farklı işlemleri veya durumları ifade ettiği durumları tanımlar. Context switch sırasında bu durum sorun yaratabilir çünkü aynı isimdeki işlemler arasında karışıklık yaşanabilir (Hennessy & Patterson, 1989, p. B-36-B-38).

4.1 Güvenlik(protection)

4.1.1 Problem

Fiziksel adresin çevrilmesi sırasında sayfa düzeyinde koruma ve koruma bilgileri kontrol edilir. Virtual cache bu çevirme işlemini yapmadığı için context switch sırasında, bir işlemin önbellekte (cache) bulunan verileri, başka bir işlemin erişimine açık hale gelebilir ve bu durumda güvenlik, bellek sızıntısı ve benzeri problemler ortaya çıkabilir (Cekleov & Dubois, 1997, p. 66; Hennessy & Patterson, 1989, p. B-36-B-38).

4.1.2 Çözüm

R ve M Bitlerinin Güncellenmesi R (Read) ve M (Modified) bitleri, bir sayfanın durumunu izlemek için kullanılır. Bu bitlerin güncellenmesi, bir sayfanın son erişim durumu ve değiştirilip değiştirilmediğini belirler. Eğer bir işlem sayfayı okuduysa R bitinin, sayfayı değiştirdiyse M bitinin güncellenmesi gerekir (Cekleov & Dubois, 1997, p. 66).

TLB Yönetimi TLB (Translation Lookaside Buffer), bellekteki sayfaların hızlı bir şekilde erişimini sağlar. TLB'de R ve M bitlerinin kopyaları yoktur çünkü bu bitlerin işlevleri, erişim hakları gibi diğer bitlerle taklit edilebilir. TLB'de herhangi bir güncelleme gerektiğinde, yazılım tuzağı (trap) kullanarak işlemci, sayfa tablosunu güncelleyebilir. Bu, güvenliğin artırılmasına yardımcı olur (Cekleov & Dubois, 1997, p. 66).

Bellek Temizliği Context switch sırasında, cache ve diğer geçici bellek alanlarının temizlenmesi, bir işlemin verilerinin başka bir işlem tarafından kullanılmasını önlemek için etkili bir yöntemdir (Hennessy & Patterson, 1989, p. B-36-B-38).

Güvenlik Protokollerinin Güncellenmesi İşletim sisteminin güvenlik açıklarını kapatmak için düzenli güncellemeler yapması, bu tür problemleri azaltabilir (Hennessy & Patterson, 1989, p. B-36-B-38).

4.2 Alias(synonyms)

4.2.1 Problem

Aynı physical address'e farklı virtual address'lerle erişilmesinden kaynaklanır. Bu durumda, virtual cache kullanan sistemler aynı physical address iki farklı virtual address ifade edildiğinde, cache'de aynı verinin iki farklı kopyası oluşabilir. Eğer bu kopyalardan biri değiştirilirse, diğer kopya güncellenmediği için veri tutarsızlığı ortaya çıkar. (Inouye et al. 1992, pp. 3–5; Hennessy & Patterson, 1989, p. B-38)

4.2.2 Çözüm

Antialiasing Bu yöntem, cache bloklarına her zaman eşsiz bir fiziksel adres atayarak bu probleme donanımsal bir çözüm sağlar. Temel çalışma prensibi; işlemci, cache miss aldığı anda o veriyle aynı physical address'e sahip başka bir synonym bulunmadığından emin olmak için tüm olası konumları kontrol eder. Eğer mevcut physical address ile eşleşen bir synonym bulunursa, bu eskisi geçersiz kılınır. Bu sayede, yeni veri önbelleğe yüklendiğinde physical adres'lerin eşsiz olması ve veri tutarlılığı sağlanır. (Hennessy & Patterson, 1989, p. B-38)

Page Coloring Page Coloring yöntemiyle aynı fiziksel adresi paylaşan sanal adreslerin belirli adres bitlerini aynı olacak şekilde ayarlanır. Bu sayede eğer birden fazla virtual adres aynı fiziksel adresi gösteriyorsa aynı konuma yerleşirler . Sonuç olarak bu yöntem yazılımsal bir çözüm olup synonyms problemini çözer ve veri tutarlılığı sağlar. (Hennessy & Patterson, 1989, p. B-38)

4.3 Cache flushing

4.3.1 Problem

Her işlem, virtual address ile verilere erişir, ancak bu adresler her işlem için farklı physical address'e karşılık gelir. Bu nedenle, yapılan iş değiştiğinde, önbellekte bulunan veriler yeni işlem için geçerli olmayan, önceki işlemin verileri olabilir. Bu durum, veri tutarsızlığına ve güvenlik risklerine yol açar. Bu nedenle, yeni işlem başlamadan önce önbellekteki verilerin tamamen temizlenmesi (flushing) gerekebilir. (Hennessy & Patterson, 1989, p. B-37-B-38)

4.3.2 Çözüm

PID Kullanımı Bu daha çok sorunu hafifletmek için kullanılan bir çözüm olup kabaca cache adres etiketlerine işlem kimlik tanımlayıcıları (PID) ekleme işlemidir. İşletim sistemi her işleme bir PID atarsa, yalnızca aynı PID tekrar kullanıldığında cache temizlenmesi gerekir. Bu yöntem, önbellekte hangi verilerin hangi işleme ait olduğunu ayırt ederek gereksiz flush işlemlerini azaltır (Cekleov & Dubois, 1997, p. 66).

4.4 Homonym

4.4.1 Problem

Homonym problemi, aynı terimin birden fazla farklı anlamı veya referansı olmasından kaynaklanan bir sorundur. Bu durum, aynı isim kullanılarak farklı değişkenlerin veya durumların ifade edilmesi anlamına gelir (Inouye et al., 1992, p. 3).

4.4.2 Çözüm

Global Address Space Homonym problemini önlemek için global bir adres alanı kullanır, böylece her işlem kendi sanal adres alanına sahiptir (Inouye et al., 1992, p. 3).

Process Tags Her sanal adresin yanında işlem etiketleri kullanılır. Böylece her işlem kendi adres alanına ve etiketine sahiptir, bu da referansların netleşmesini sağlar (Inouye et al., 1992, p. 3).

DEĞERLENDİRME

Bu araştırma ile virtual memory kullanımı sırasında context switch'in virtual cache üzerindeki etkilerini ve olası sorunlarını inceledim. Virtual memory, belleğin daha verimli kullanılması ve aynı anda birden fazla işlemin yapılabilmesi için belleğin bir bölümünü gerektiğinde RAM olarak kullanmak üzere ayıran bir tekniktir. Bu işlemler aynı fiziksel belleği kullandığından bazı problemler ortaya çıkmaktadır. Cache, RAM'e doğrudan erişmek yerine sık kullanılan verilerin tutulduğu küçük ama hızlı bir bellek türüdür. Bu sistemin amacı, verilere daha hızlı ulaşarak performansı artırmaktır. Bellekteki verilere ulaşmak için sanal adreslerin fiziksel adreslere dönüştürülmesi gerekir ve bu durum, iki farklı cache türünün ortaya çıkmasına sebep olmuştur. Birincisi, fiziksel adresleri kullanan fiziksel cache (physical cache); ikincisi ise sanal adresleri kullanan sanal cache (virtual cache) olarak bilinir. Fiziksel cache'ler daha güvenilir olmasına rağmen, virtual cache'ler özellikle çoklu görev yapma ve performans arttırımı gibi avantajlar sundukları için tercih edilir. Fakat virtual cache'in bazı problemleri vardır: güvenlik problemi, alias problemi, I/O cihazlarının adreslerinin dönüştürülmesi sırasında yaşanan problemler ve context switch problemleri. Bu problemlerin bazıları birbirlerini kapsamakta olup, flushing gibi problemler aynı zamanda cache yapısının bir parçası ve bir çözüm yoludur.

Araştırmamda, context switch durumunda oluşan problemler ve bunların çözümüne yoğunlaştım; bu sırada güvenlik, alias, homonym ve veri tutarsızlığı problemlerini inceledim. Güvenlik problemi, aslında fiziksel adresten sanal adrese dönüşüm sırasında yapılması gereken güvenlik kontrollerinin virtual önbellekte yapılmaması ve bir hata durumu olmadıkça kontrol edilmemesinden kaynaklanır. Bu bir işlem için önbellekte bulunan verilerin başka bir işlem tarafından erişilmesine olanak tanır ve bellek sızıntılarına sebep olur. Bunu önlemek için R ve M bitleri, TLB yöntemi, bellek temizliği ve güvenlik protokollerinin daha çok kontrol edilmesi gibi yöntemler tercih edilebilir. Alias problemi, aynı fiziksel adrese birden fazla sanal adresin işaret etmesi durumudur. Bunu önlemek için yazılımsal ve donanımsal çözümler geliştirilmiştir; donanımsal çözüm olarak anti-aliasing, yazılımsal çözüm olarak ise page coloring yöntemi kullanılır. Anti-aliasing, cache bloklarına her zaman benzersiz bir adres verir ve aynı fiziksel adrese sahip başka bir veri olmadığından emin olur. Page coloring ise aynı fiziksel adresleri paylaşan sanal adreslerin belli bitlerinin aynı olmasını sağlar. Üçüncü problem ise cache flushing problemidir. Cache flushing, bir cache belleğin içindeki tüm verilerin temizlenmesi ve belleğin sıfırlanması işlemidir. Bu işlem genellikle yeni verilerin yüklenmesi, uygulamalar arası geçiş veya güvenlik nedeniyle gerçekleştirilir. Problem oluşturabileceği nokta, context switch sonrasında önbellekte bulunan verilerin önceki işlemin verileri olup yanlış verilerin kullanılmasına sebep olabilmesidir. Bunun çözümü ise cache adres etiketlerine PID eklemektir. Son problemimiz homonym problemidir. Bu, aynı ifadenin birden fazla fiziksel adresi göstermesinden kaynaklanan bir problemdir ve Global Address Space ve Process tag kullanımı ile önlenabilir.

KAYNAKÇA

- Cekleov, M., & Dubois, M. (1997). Virtual-address Caches: Part 1 – Problems and solutions in uniprocessors [Digital]. *IEEE Micro*, 17(5), 64-71. <http://classweb.ece.umd.edu/enee646.F2007/Cekleov1.pdf>
- Hennessy, J. L., & Patterson, D. A. (1989). *Computer Architecture: a Quantitative approach* [Digital]. [https://acs.pub.ro/~cpop/SMPA/Computer%20Architecture%20A%20Quantitative%20Approach%20\(5th%20edition\).pdf](https://acs.pub.ro/~cpop/SMPA/Computer%20Architecture%20A%20Quantitative%20Approach%20(5th%20edition).pdf)
- Inouye, J., Konuru, R., Walpole, J., & Sears, B. (1992). The effects of virtually addressed caches on virtual memory design and performance [Digital]. *ACM SIGOPS Operating Systems Review*, 26(4), 14-29. <https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://digitalcollections.ohsu.edu/record/3954/files/csetech-271.pdf&ved=2ahUKEwjH-J3znLGJAXXOcFEDHXNQC4UQFnoECBQQAQ&usg=AOvVaw1W0AZoHVrsVqMVQderDjm8>
- Silberschatz, A., Galvin, P. B., & Gagne, G. (1983). *Operating system concepts* [Digital]. <https://archive.org/details/operatingsystemconcepts10th/page/n445/mode/2up>