GAZİ ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BM311 BİLGİSAYAR MİMARİSİ ÖDEVİ I

Öğrenci Adı – Soyadı:	Said Berk
Öğrenci Numarası:	21118080070
Ders Kodu/Adı:	BM311/Bilgisayar Mimarisi
Teslim Tarihi:	28.10.2024

Araştırma

1. Problem

Virtual memory kullanıldığında uygulamalar arasında geçişte virtual cache ile karşılaşılan problemler ve kullanılan çözümlerin araştırılması ve uygun bir formatta raporlaştırılması.

2. Kavramlar

Araştırmanın doğru bir şekilde anlaşılmasını sağlamak için bazı kavramların bu ödevde nasıl değerlendirildiğinin belirtilmesi gerekli görülmüştür.

2.1 Fiziksel Adres (Phsyical Memory Adress)

Bilgisayarın herhangi bir hafıza donanımı üzerindeki erişilebilir bölgeleri birbirinden ayırt etmek için kullanılan, somut bir bölgeyi işaret eden global belirteç. (Intel Corporation, 2024, Section 3.3)

2.2 Sanal Adres (Virtual Memory Adress)

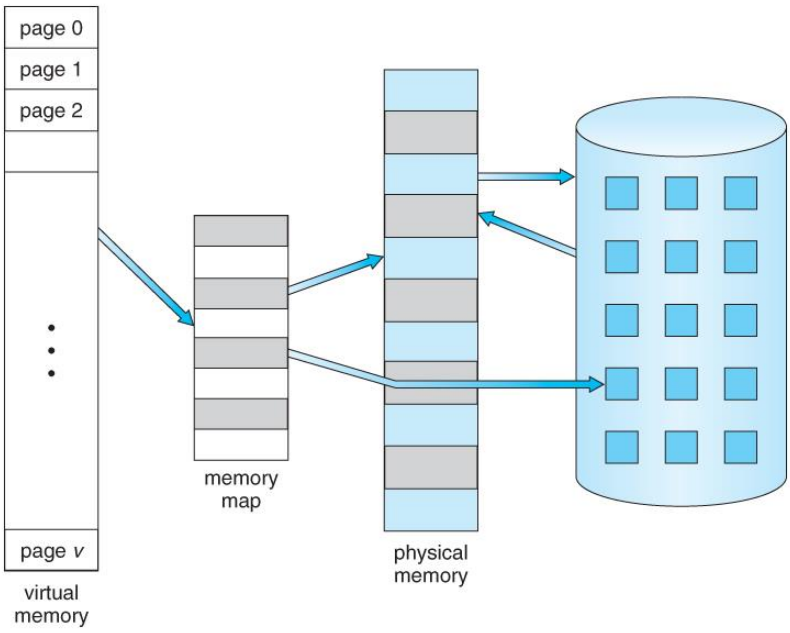
İşletim sistemi tarafından yönetilen adres uzayıdır, bellek sayfalama sistemleri tarafından *referans tablo* ile kullanılır. *(Intel Corporation, 2024, Section 3.3.2)*

2.3 Mantıksal Adres (Logical Memory Adress)

Performans ve alan verimliliği kaygısıyla oluşturulmuş, fiziksel adresin encode edilerek kısaltılmış hali. *(Intel Corporation, 2024, Section 3.4)*

2.4 Sanal Önbellek (Virtual Cache)

Herhangi bir fiziksel adrese çevirme işlemi olmaksızın doğrudan sanal adreslerle önbellek faaliyetlerini yürütebilen önbellek türü. *(Hennessy & Patterson, 2021, Appendix B)*



Abraham Silberschatz, Greg Gagne, and Peter Baer Galvin, "Operating System Concepts, Ninth Edition ", Chapter 9

3. Metotlar

Araştırmada kullanılan metotlar şu şekilde listelenebilir:

3.1 Ders Kitabı Taraması

BM311 Bilgisayar Mimarisi dersinin https://bigdata.gazi.edu.tr/akcayol/BM311.htm adresinde belirtilen ana ders kitabı *Stallings, W., "Computer Organization and Architecture 11/e", Pearson, 2021.* ve yardımcı kaynaklar başlığı altında belirilen *Hennessy, J.L., Patterson, D.A., "Computer Architecture a Quantitative Approach 6/e", Morgan Kaufmann, 2019.* Kitaplarının araştırma için taranması.

3.2 Geliştirici El Kitapları Taraması

Intel® 64 and IA-32 Architectures Software Developer Manuals kaynağındaki talimatlara güven duyulması ve örneklerde bahsedilmesi.

3.3 Görsel Materyal Taraması

Araştırılan konunun okuyucuya daha iyi aktarılabilmesi için, rapora yerleştirmek amacıyla güvenilir kaynaklardan görsellerin alıntılanması.

3.4 Doğal Dil İşleme Modellerinin Bilgi Organizasyonunda Kullanılması

OpenAI ChatGPT 4o ve o1 modelleriyle birlikte *Anthropic Claude Sonnet 3.5* büyük dil modellerinin araştırma verisi toplama, kaynak önerme ve veri sınıflandırma işlerinde kullanılması.

4. Virtual Cache ile İlgili Karşılaşılan Problemler

"Yaygın durumun hızlı hale getirilmesi kılavuzu, önbellek için *sanal adreslerin* kullanılmasını önerir, çünkü isabetler, kaçırmalardan çok daha yaygındır. Bu tür önbelleklere *sanal önbellekler* denir, fiziksel adresleri kullanan geleneksel önbelleği tanımlamak için ise fiziksel önbellek terimi kullanılır." (Hennessy & Patterson, 2012, s. B-35)

4.1 Güvenlik Problemleri

Önbellek sanallaştırması, sanal adreslerin fiziksel adrese dönüştürülmesi aşamasını gerektirdiği için bu aşamada çeşitli güvenlik sorunlarına yol açar. (Hennessy & Patterson, 2012, s. B-35)

4.1.1 Hatalı Erişim ve Bellek Yalıtımı Sorunları

Sanal adresler doğru şekilde manipüle edilirse önbellekteki diğer kullanıcıların fiziksel adreslerini çözümleme ve verilerine erişim mümkün kılınır. Bu noktada önlemler alınması gerekir. (Garfinkel & Rosenblum)

4.1.2 Sayfa Tabanlı Koruma Eksikliği

Adres çevirisi önbellek erişiminden önce olmadığı için sanal önbellekler sayfa tabanlı güvenlik önlemlerinin dışında kalabilir. Bu durum, bellek erişimin izinlerinin kontrol mekanizmasının bypass edilmesine sebep olabilir. (Garfinkel & Rosenblum)

4.2 Güvenlik Problemleri Önlemleri

4.2.1 Hatalı Erişim ve Bellek Yalıtımı Sorunları Önlemleri

4.1. Başlıkta önerilen iki probleme geliştirilen önlemler aşağıdaki gibi listelenebilir:

4.2.1.1 Erişim Kontrol Listeleri (ACLs):

Garfinkel ve Rosenblum'un önerdiği şekilde her bir bellek sayfası için erişim izinlerini belirlemek ve bu izinleri sıkı bir şekilde denetlemek, yetkisiz erişimleri önlemeye yardımcı olabilir. (Garfinkel & Rosenblum)

4.2.1.2 Güvenli Bellek Yönetimi:

Garfinkel ve Rosenblum'un önermesi olan bellek tahsisi ve serbest bırakma işlemlerinin güvenli bir şekilde yapılması, bellek sızıntılarını ve hatalı erişimleri önleyebilir. (Garfinkel & Rosenblum)

4.2.2 Hatalı Erişim ve Bellek Yalıtımı Sorunları Önlemleri

4.2. Başlıkta önerilen iki probleme geliştirilen önlemler aşağıdaki gibi listelenebilir:

4.2.2.1 Donanım Destekli Güvenlik Özellikleri:

Modern işlemciler, bellek korumasını donanım seviyesinde destekleyen özellikler sunar. Örneğin, Intel'in VT-x ve AMD'nin AMD-V teknolojileri, sanal makineler için donanım tabanlı yalıtım sağlar.. *(Michael Pearce, Sherali Zeadally, & Ray Hunt)*

4.2.2.2 Gelişmiş TLB Yönetimi:

Çeviri Arabellek (TLB) yönetimini optimize etmek ve koruma bilgilerini TLB ile entegre etmek, sanal önbelleklerin sayfa tabanlı koruma mekanizmalarından faydalanmasını sağlayabilir. *(Hennessy & Patterson, 2012, s. B-35)*

4.2.2.3 Yazılım Tabanlı İzleme ve Yönetim:

Yazılım tabanlı güvenlik araçları, bellek erişimlerini izleyebilir ve anormal davranışları tespit ederek müdahale edebilir. *(Michael Pearce, Sherali Zeadally, & Ray Hunt)*

4.3 Önbellek Bağlam Anahtarlama Problemi

Her proccess değişikliğinde, instruction ve data fiziksel adresleri de değişeceğinden, yönlendirilen fiziksel adreslerin de değiştirilmesi gerekmektedir. Bu gerekliliğin sağlanması için, her uygulama arası geçişte önbelleğin tekrar tekrar resetlenmesi gerekmektedir. Bu durum, fazladan bir hesaplama ve zaman maliyeti doğurmaktadır. *(Hennessy & Patterson, 2012, s. B-36)*

4.4 Önbellek Bağlam Anahtarlama Problemi Önlemi

Önbellek adres etiketinin genişliğini bir işlem tanımlayıcı etiket (PID) ile artırmak önlem kabul edilir. İşletim sistemi bu etiketleri işlemlere atarsa, yalnızca bir PID yeniden kullanıldığında önbelleği temizlemesi gerekir; yani, PID, önbellekteki verilerin ilgili işlem için olup olmadığını ayırt etme yeteneğine erişir. *(Hennessy & Patterson, 2012, s. B-36)*

4.5 Aliasing & Synonym Problemi

Sanal önbellek map stratejisine göre bir sanal önbellek için birden fazla sanal adres aynı fiziksel adresi tanımlamak için kullanılabilir. *(Hennessy & Patterson, 2012, s. B-37)*

"Bu tür yinelenen adresler, *'synonyms'* veya *'aliases'* olarak bilinir ve sanal bir önbellekte aynı verinin iki farklı kopyasının bulunmasına yol açabilir. Eğer bu kopyalardan biri değiştirilirse, diğer kopya yanlış bir değer içerebilir. Ancak, fiziksel bir önbellekte bu sorun yaşanmaz çünkü erişimler önce aynı fiziksel adres bloğuna dönüştürülür." *(Hennessy & Patterson, 2012, s. B-35)*

i	Önbellek satır numarası
j	Main memory blok numarası
m	Önbellekteki satır sayısı

Olmak üzere sanal makine indisi hesaplama formülü

$$i = j \% m$$

Olarak verilebilir. (Akçayol, 2024). Bu işlem sonucunda j ve m'nin aynı değerlere sahip olduğu bütün i değerleri için aynı indis hesaplanır ve adres yinelemesi oluşur.

4.6 Aliasing & Eş Anlamlılık Problemi Önlemi

4.6.1 Donanım Çözümü

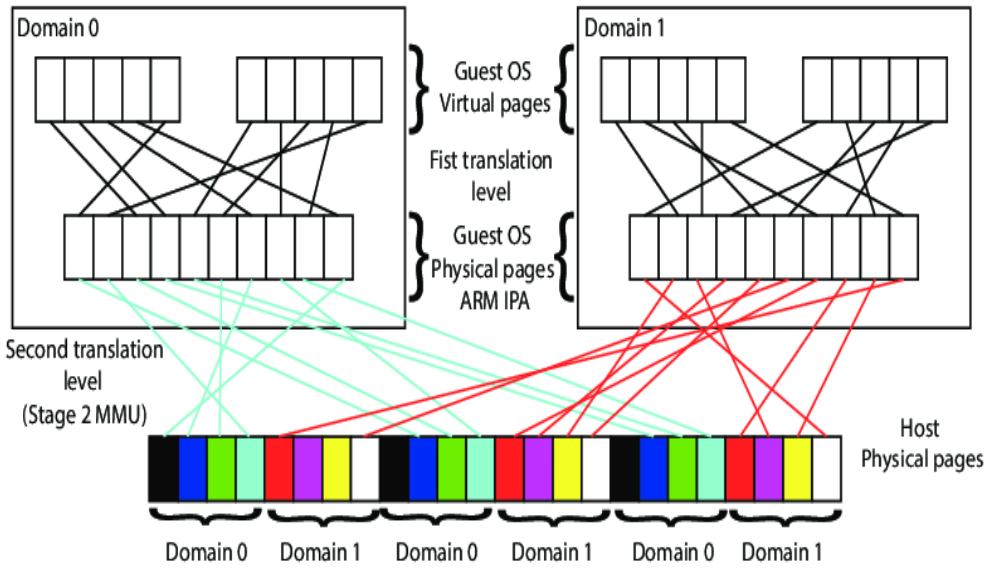
Eş anlamlılık (alias) problemine yönelik donanım çözümleri, antialiasing olarak adlandırılır ve her önbellek bloğuna benzersiz bir fiziksel adres garanti eder. Örneğin, AMD Opteron, 4 KB sayfa boyutuna ve iki yollu set ilişkisine sahip 64 KB'lık bir komut önbelleği kullanır; bu nedenle, donanım, set indeksindeki üç sanal adres bitiyle ilgili eş anlamlılıkları yönetmelidir. Eş anlamlılıkları önlemek için, bir hata durumunda dört setin her birinde iki blok olmak üzere tüm sekiz olası konumu kontrol ederek, getirilen verinin fiziksel adresiyle eşleşen bir adres olmadığından emin olur. Eğer bir eşleşme bulunursa, bu blok geçersiz kılınır, böylece yeni veriler önbelleğe yüklendiğinde fiziksel adreslerinin benzersiz olması sağlanır. *(Hennessy & Patterson, 2012, s. B-38)*

4.6.2 Yazılım Çözümü (Sayfa Renklendirme Yöntemi)

- Her sayfaya bir "renk" atanır
- Bu renk, sayfanın fiziksel adresinin belirli bitlerinden belirlenir
- Aynı renkteki sayfalar, önbellekte aynı setlere eşlenir

$$Fiziksel\ Adres: [Sayfa\ Numarası]\ [Sayfa\ Offset]$$

- Sayfa numarasının alt bitleri "renk bitleri" olarak kullanılır
- Bu bitler, önbellek setlerini belirler
- İşletim sistemi, sayfa tahsisi yaparken bu renkleri dikkate alır.



Modica, P., Biondi, A., Buttazzo, G., & Patel, A. (2018). Supporting temporal and spatial isolation in a hypervisor for ARM multicore platforms. In *2018 IEEE International Conference on Industrial Technology (ICIT)* (pp. 1651-1657). IEEE. https://doi.org/10.1109/ICIT.2018.8352429

Değerlendirme

Önbellekler, verinin sürekli ve hızlıca erişilmesi gereken durumlar için tasarlanmış başarılı bir mühendislik örneği. Sezgisel olarak neden daha hızlı veriye eriştiklerini anlamak da fazlasıyla mümkün. Günlük hayatta hızlıca bir veriye erişmek istediğimizde uyguladıklarımızla yöntem konsept olarak aynı.

Örneğin, sürekli faydalandığım fakat aklımda tutamayacağım bir veri varsa – bu akılda zor tutulacak bir telefon numarası veya IBAN numarası olabilir – bunu bir kenara not alıyorsam önbelleklerin çalışma mantığını da bundan farklı düşünmemek gerek. RAM ile kıyasla daha hızlı olmasının sebebi de aynı şekilde fazlasıyla anlaşılır durumda, mesafe ve bağlanma şekilleri RAM'e göre daha kısa ve daha az karmaşık. Bütün bu sebepler önbelleği sistemin vazgeçilmez bir parçası yapıyor.

Bu avantajlı durumu ana bellekteki bütün veriler için kullanmak mümkün değil. Ortalama olarak ana belleğin $1.6\ x\ 10^{-5}$ 'i kadar küçük yazılabilir/okunabilir hafızaya sahip bir donanım olan önbellekte bütün programların bağlamını (programla ilgili komut ve verilerin) tutmak mümkün değil. Bu sebeple o an mevcut çalışan uygulamaların çalışma bağlamı için önbellekten faydalanmak doğru bir yaklaşım olacaktır fakat aynı anda birden fazla bağlamın dar bir ortam olan önbellekte olmasının *birtakım zararlı durumlara yol açtığı yukarıda tartışılmıştır.*

Bunlardan en başlıcası ve en kritiği güvenlik endişeleridir. Sahiplikleri farklı olan verilerin yan yana saklanmak zorunluluğu kötü niyetli veya kaza sonucu oluşabilecek izinsiz veri erişimlerine ortam sağlamakta ve bu da asıl sorumuz olan *virtual memory kullanıldığında uygulamalar arasında geçişte*

virtual cache ile karşılaşılan problemlerin kendi kanaatimce sonuçları en ciddi olan problemlerinden birisi.

Güvenlik problemlerinin yegane sebebi, kötü bellek yalıtımı ve yönetimi. Bellek yalıtımı uygulamaları ise başlıca güvenlik önlemi. Bellek yalıtımı ise hem yazılımsal hem de donanımsal yöntemlerle mümkün.

Güvenlik problemleri genel olarak bütün önbelleklerin problemiyken aliasing (eş anlamlılık problemi) araştırma konusu olan yalnızca virtual memory'e özgü problemlerden birisi olarak anlaşılmakta. Sebebi ise mantıksal adresleri kendi algoritmasına göre fiziksel adresleri encode etmesinde kullanmasında. Bu encode kısmında kendisinden ***çok daha geniş bir adres uzayına sahip ana belleği daraltmaya çalışmasından ötürü belli hücrelere birden fazla veri bloğu yazılabiliyor*** ve bu durum farklı blokların aynı isme sahip olmasına neden oluyor. Sanal bellek kullanımındaki en sık karşılaşılan problemlerden birisi de araştırma sonucu olarak aliasing problemi olarak anlaşılıyor.

Asıl sorun olan eş anlamlılık probleminin de tıpkı güvenlik probleminde olduğu gibi hem yazılımsal hem de donanımsal çözümleri var. Ek donanım desteğiyle fiziksel olarak çözüme kavuşturulabilen bu sorun ***Sayfa Renklendirme Algoritması*** yardımıyla yazılımsal olarak da bir çözüme çıkabiliyor. Bu tarz mimari problemlerinin hem yazılımsal hem de donanımsal çözümleri olması ufuk açıcı bir tecrübe olarak araştırma kazanımlarına geçiyor.

Sonuç olarak, fiziksel önbelleği sanallaştırma pek çok avantajla gelse de baş edilmesi gereken bir karmaşıklıkla, güvenlikle ve encode problemleriyle birlikte bu sorunları sağlıyor. Bu sorunların farkında olunması, CPU için veri yönetimi konusunda hayati öneme sahip.

REFERANSLAR

1) Intel Corporation. (2024). Intel® 64 and IA-32 architectures software developer's manual: Vol. 1. Basic architecture (Order No. 253665-074US). https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html

2) Hennessy, J. L., & Patterson, D. A. (2012). Computer architecture: A quantitative approach (5. baskı). Morgan Kaufmann.

3) Silberschatz, A., Gagne, G., & Galvin, P. B. (2013). Operating system concepts (9th ed.). Wiley.

4) Garfinkel, T., & Rosenblum, M. (2005). When virtual is harder than real: Security challenges in virtual machine based computing environments. Stanford University Department of Computer Science.

5) Pearce, M., Zeadally, S., & Hunt, R. (2013). Virtualization: Issues, security threats, and solutions. *ACM Computing Surveys, 45*(2), Article 17. https://doi.org/10.1145/2431211.2431216

6) Akçayol, M. A. (2024). *BM311 Bilgisayar Mimarisi Dersi* [Lecture slides]. Retrieved October 28, 2024, from https://bigdata.gazi.edu.tr/akcayol/BM311.htm

7) Modica, P., Biondi, A., Buttazzo, G., & Patel, A. (2018). Supporting temporal and spatial isolation in a hypervisor for ARM multicore platforms. In *2018 IEEE International Conference on Industrial Technology (ICIT)* (pp. 1651-1657). IEEE. https://doi.org/10.1109/ICIT.2018.8352429