



**Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği Bilgisayar
Organizasyonu**

Ödev:4

Dersin Yürütücüsü:Dr.Öğr.Üyesi Erkan Uslu

ÖĞRENCİ NUMARASI:19011095

AD:EFE

SOYAD:GİRGİN

DIMENSION SABİTİ:64

```
valgrind --tool=cachegrind ./1.out
```

```
==8525== Cachegrind, a cache and branch-prediction profiler
```

```
==8525== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
```

```
==8525== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
```

```
==8525== Command: ./1.out
```

```
==8525==
```

```
--8525-- warning: L3 cache found, using its data for the LL simulation.
```

```
secs:0.115834
```

```
==8525==
```

```
==8525== I refs: 12,934,856
```

```
==8525== I1 misses: 1,477
```

```
==8525== LLi misses: 1,451
```

```
==8525== I1 miss rate: 0.01%
```

```
==8525== LLi miss rate: 0.01%
```

```
==8525==
```

```
==8525== D refs: 5,058,320 (4,708,176 rd + 350,144 wr)
```

```
==8525== D1 misses: 52,715 ( 50,518 rd + 2,197 wr)
```

```
==8525== LLd misses: 3,469 ( 1,340 rd + 2,129 wr)
```

```
==8525== D1 miss rate: 1.0% ( 1.1% + 0.6% )
```

```
==8525== LLd miss rate: 0.1% ( 0.0% + 0.6% )
```

```
==8525==
```

```
==8525== LL refs: 54,192 ( 51,995 rd + 2,197 wr)
```

```
==8525== LL misses: 4,920 ( 2,791 rd + 2,129 wr)
```

```
==8525== LL miss rate: 0.0% ( 0.0% + 0.6% )
```

```
valgrind --tool=cachegrind ./2.out
```

```
==8660== Cachegrind, a cache and branch-prediction profiler
```

```
==8660== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
```

```
==8660== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
```

```
==8660== Command: ./2.out
```

```
==8660==
```

```
--8660-- warning: L3 cache found, using its data for the LL simulation.
```

```
secs:0.114649
```

```
==8660==
```

```
==8660== I refs: 12,934,849
```

```
==8660== I1 misses: 1,473
```

```
==8660== LLi misses: 1,448
```

```
==8660== I1 miss rate: 0.01%
```

```
==8660== LLi miss rate: 0.01%
```

```
==8660==
```

```
==8660== D refs: 5,058,318 (4,708,175 rd + 350,143 wr)
```

```
==8660== D1 misses: 52,715 ( 50,518 rd + 2,197 wr)
```

```
==8660== LLd misses: 3,469 ( 1,340 rd + 2,129 wr)
```

```
==8660== D1 miss rate: 1.0% ( 1.1% + 0.6% )
```

```
==8660== LLd miss rate: 0.1% ( 0.0% + 0.6% )
```

```
==8660==
```

```
==8660== LL refs: 54,188 ( 51,991 rd + 2,197 wr)
```

```
==8660== LL misses: 4,917 ( 2,788 rd + 2,129 wr)
```

```
==8660== LL miss rate: 0.0% ( 0.0% + 0.6% )
```

```
valgrind --tool=cachegrind ./3.out
```

```
==8787== Cachegrind, a cache and branch-prediction profiler
```

```
==8787== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
```

```
==8787== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
```

```
==8787== Command: ./3.out
```

```
==8787==
```

```
--8787-- warning: L3 cache found, using its data for the LL simulation.
```

```
secs:0.116057
```

```
==8787==
```

```
==8787== I refs: 12,934,856
```

```
==8787== I1 misses: 1,477
```

```
==8787== LLi misses: 1,451
```

```
==8787== I1 miss rate: 0.01%
```

```
==8787== LLi miss rate: 0.01%
```

```
==8787==
```

```
==8787== D refs: 5,058,320 (4,708,176 rd + 350,144 wr)
```

```
==8787== D1 misses: 52,715 ( 50,518 rd + 2,197 wr)
```

```
==8787== LLd misses: 3,469 ( 1,340 rd + 2,129 wr)
```

```
==8787== D1 miss rate: 1.0% ( 1.1% + 0.6% )
```

```
==8787== LLd miss rate: 0.1% ( 0.0% + 0.6% )
```

```
==8787==
```

```
==8787== LL refs: 54,192 ( 51,995 rd + 2,197 wr)
```

```
==8787== LL misses: 4,920 ( 2,791 rd + 2,129 wr)
```

```
==8787== LL miss rate: 0.0% ( 0.0% + 0.6% )
```

DIMENSION SABİTİ:256

valgrind --tool=cachegrind ./1.out

==6630== Cachegrind, a cache and branch-prediction profiler

==6630== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.

==6630== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info

==6630== Command: ./1.out

==6630==

--6630-- warning: L3 cache found, using its data for the LL simulation.

secs:7.464452

==6630==

==6630== I refs: 783,306,031

==6630== I1 misses: 1,445

==6630== L1i misses: 1,420

==6630== I1 miss rate: 0.00%

==6630== L1i miss rate: 0.00%

==6630==

==6630== D refs: 306,631,868 (288,660,378 rd + 17,971,490 wr)

==6630== D1 misses: 16,894,756 (16,869,520 rd + 25,236 wr)

==6630== LLd misses: 26,505 (1,334 rd + 25,171 wr)

==6630== D1 miss rate: 5.5% (5.8% + 0.1%)

==6630== LLd miss rate: 0.0% (0.0% + 0.1%)

==6630==

==6630== LL refs: 16,896,201 (16,870,965 rd + 25,236 wr)

==6630== LL misses: 27,925 (2,754 rd + 25,171 wr)

==6630== LL miss rate: 0.0% (0.0% + 0.1%)

```
valgrind --tool=cachegrind ./2.out
```

```
==7288== Cachegrind, a cache and branch-prediction profiler
```

```
==7288== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
```

```
==7288== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
```

```
==7288== Command: ./2.out
```

```
==7288==
```

```
--7288-- warning: L3 cache found, using its data for the LL simulation.
```

```
secs:7.372939
```

```
==7288==
```

```
==7288== I refs: 783,306,020
```

```
==7288== I1 misses: 1,445
```

```
==7288== LLi misses: 1,420
```

```
==7288== I1 miss rate: 0.00%
```

```
==7288== LLi miss rate: 0.00%
```

```
==7288==
```

```
==7288== D refs: 306,631,864 (288,660,374 rd + 17,971,490 wr)
```

```
==7288== D1 misses: 2,140,708 ( 2,115,472 rd + 25,236 wr)
```

```
==7288== LLd misses: 26,505 ( 1,334 rd + 25,171 wr)
```

```
==7288== D1 miss rate: 0.7% ( 0.7% + 0.1% )
```

```
==7288== LLd miss rate: 0.0% ( 0.0% + 0.1% )
```

```
==7288==
```

```
==7288== LL refs: 2,142,153 ( 2,116,917 rd + 25,236 wr)
```

```
==7288== LL misses: 27,925 ( 2,754 rd + 25,171 wr)
```

```
==7288== LL miss rate: 0.0% ( 0.0% + 0.1% )
```

```
valgrind --tool=cachegrind ./3.out
```

```
==7801== Cachegrind, a cache and branch-prediction profiler
```

```
==7801== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
```

```
==7801== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
```

```
==7801== Command: ./3.out
```

```
==7801==
```

```
--7801-- warning: L3 cache found, using its data for the LL simulation.
```

```
secs:7.587796
```

```
==7801==
```

```
==7801== I refs: 783,306,038
```

```
==7801== I1 misses: 1,449
```

```
==7801== L1i misses: 1,423
```

```
==7801== I1 miss rate: 0.00%
```

```
==7801== L1i miss rate: 0.00%
```

```
==7801==
```

```
==7801== D refs: 306,631,870 (288,660,379 rd + 17,971,491 wr)
```

```
==7801== D1 misses: 18,967,074 ( 18,941,838 rd + 25,236 wr)
```

```
==7801== LLd misses: 26,505 ( 1,334 rd + 25,171 wr)
```

```
==7801== D1 miss rate: 6.2% ( 6.6% + 0.1% )
```

```
==7801== LLd miss rate: 0.0% ( 0.0% + 0.1% )
```

```
==7801==
```

```
==7801== LL refs: 18,968,523 ( 18,943,287 rd + 25,236 wr)
```

```
==7801== LL misses: 27,928 ( 2,757 rd + 25,171 wr)
```

```
==7801== LL miss rate: 0.0% ( 0.0% + 0.1% )
```

Cache details				
Cache:	L1 data	L1 instruction	L2	L3
Size:	4 x 32 KB	4 x 32 KB	4 x 256 KB	8 MB
Associativity:	8-way set associative	8-way set associative	4-way set associative	16-way set associative
Line size:	64 bytes	64 bytes	64 bytes	64 bytes
Comments:	Direct-mapped	Direct-mapped	Non-inclusive Direct-mapped	Inclusive Shared between all cores

Level 1 cache size ?	4 x 32 KB instruction caches 4 x 32 KB data caches	▼
Level 2 cache size ?	4 x 256 KB	▼
Level 3 cache size	8 MB	▼

Ayarlar

efe girgin
Efegirgin05@outlook.com

Bir ayar bulun

Giriş

Sistem

Bluetooth ve cihazlar

Ağ ve internet

Kişiselleştirme

Uygulamalar

Hesaplar

Zaman ve dil

Oyun

Erişilebilirlik

Gizlilik ve güvenlik

Windows Update

Sistem > Sistem bilgisi

DESKTOP-2UBV9SP
ABRA AS V16.4

Bu bilgisayarı yeniden adlandır

Cihaz özellikleri

Cihaz adı
DESKTOP-2UBV9SP

İşlemci
Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz 2.40 GHz

Takılı RAM
16,0 GB (kullanılabilir: 15,8 GB)

Cihaz Kimliği
572856BF-828D-41C6-9D4A-B656DCD2CC92

Ürün Kimliği
00331-10000-00001-AA619

Sistem türü
64 bit işletim sistemi, x64 tabanlı işlemci

Kalem ve dokunma
Bu görüntü biriminde kalem girdisi veya dokunarak giriş yok

İlgili bağlantılar

Etki alanı veya çalışma grubu

Sistem koruması

Gelişmiş sistem ayarları

Windows özellikleri

Edisyon
Windows 11 Pro

Sürüm
23H2

Yüklenme tarihi
18.11.2022

İS derlemesi
22631.3447

Deneyim
Windows Feature Experience Pack 1000.22688.1000.0

Microsoft Hizmet Sözleşmesi

Microsoft Yazılım Lisans Koşulları

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

Verilen tabloya dayanarak, L1 veri ve komut önbelleklerinin hit ve miss oranlarını teorik olarak hesaplayabiliriz.

Cache Detayları:

- L1 Data Cache (L1 veri önbelleği):
- Boyut: 4 x 32 KB (her çekirdek için 32 KB)
- Associativity: 8-yollu set associative
- Satır boyutu: 64 byte
- Haritalama: Direct-mapped

- L1 Instruction Cache (L1 komut önbelleği):
- Boyut: 4 x 32 KB (her çekirdek için 32 KB)
- Associativity: 8-yollu set associative
- Satır boyutu: 64 byte
- Haritalama: Direct-mapped

Teorik Hesaplamalar

Adım 1: Toplam satır sayısını hesaplayın.

L1 veri ve komut önbelleklerinin her biri için toplam satır sayısını hesaplamalıyız:

- Her çekirdek için önbellek boyutu: 32 KB
- Satır boyutu: 64 byte
- Toplam satır sayısı = (Önbellek boyutu) / (Satır boyutu) = (32 KB) / (64 byte) = 512 satır

Adım 2: Set sayısını hesaplayın.

8-yollu set associative olduğundan, her set 8 satır içerir:

- Toplam set sayısı = (Toplam satır sayısı) / (Associativity) = 512 / 8 = 64 set

Adım 3: Teorik hit ve miss oranlarını hesaplayın.

Önbelleğin başlangıçta boş olduğunu varsayarsak, her yeni veri veya komut isteği bir miss oluşturacaktır. Zamanla, önbellek doldukça ve aynı veriler tekrar erişildikçe, hit oranı artacaktır. Ancak, başlangıçtaki senaryoda:

- İlk 512 erişimde, her erişim bir miss olacaktır (çünkü önbellek boş).
- 513. erişimden itibaren, daha önce erişilen veriler tekrar istenirse, hit oranı artmaya başlayacaktır.

Miss oranı başlangıçta: %100** (ilk 512 erişimde)

Hit oranı başlangıçta: %0 (ilk 512 erişimde)

İlerleyen erişimlerde:

- 513. erişimde, önceki 512 veriden birine erişilirse, hit oranı %1 olacaktır.
- 1024. erişimde, önceki 512 veriden birine erişilirse, hit oranı %50 olacaktır.

Özetle, önbellek başlangıçta boş olduğunda, ilk 512 erişimde miss oranı %100 olacaktır. Daha sonra, erişilen verilerin tekrar istenmesi durumunda hit oranı artacak ve miss oranı düşecektir.

Bu teorik hesaplamaları doğrulamak için, Valgrind gibi bir araç kullanarak gerçek performans verilerini elde edebilir ve karşılaştırabiliriz. Valgrind ile programın önbellek hit ve miss oranlarını ölçerek, teorik hesaplamamızın doğruluğunu test edebiliriz.

Teorik Hesaplamalar ve Valgrind Çıktılarının Karşılaştırılması

Teorik Hesaplamalar:*

Başlangıçta boş bir önbellekle başlarsak, ilk 512 erişimin (satır sayısı) tamamı miss olacaktır:

- *Miss Oranı:* %100 (ilk 512 erişim için)
- *Hit Oranı:* %0 (ilk 512 erişim için)

Sonrasında, önbellekte bulunan verilerin tekrar erişimi ile hit oranı artmaya başlar.

Valgrind Çıktıları:

64 Değeri için:

- Komut İstatistikleri:
 - I refs: 12,934,856
 - I1 misses: 1,477
 - I1 miss rate: %0.01

- Veri İstatistikleri:

- D refs: 5,058,320
- D1 misses: 52,715
- D1 miss rate: %1.0

256 Değeri için:

- Komut İstatistikleri:

- I refs: 783,306,038
- I1 misses: 1,445
- I1 miss rate: %0.00

- Veri İstatistikleri:

- D refs: 306,631,868
- D1 misses: 16,894,756
- D1 miss rate: %5.5

Analiz ve Karşılaştırma:

1. Komut İstatistikleri:

- Teorik hit oranı:* Başlangıçta %0, zamanla artar.
- Valgrind hit oranı:* %99.99 (64 değeri için), %100 (256 değeri için).

Komutlar için valgrind çıktılarında gözlemlenen hit oranı oldukça yüksektir, bu da başlangıçtaki miss oranının oldukça düşük olduğunu göstermektedir. Bu, teorik hesaplamalarımızla uyumludur; başlangıçta miss oranı yüksek olsa da, ilerleyen erişimlerde hit oranı artmaktadır.

2. Veri İstatistikleri:

- Teorik hit oranı: Başlangıçta %0, zamanla artar.
- Valgrind hit oranı: %99 (64 değeri için), %94.5 (256 değeri için).

Veriler için valgrind çıktılarında gözlemlenen hit oranları, teorik olarak beklediğimizden daha düşüktür. Bu, veri erişimlerinin çeşitliliği ve tekrarlanma sıklığı gibi faktörlere bağlı olabilir. Valgrind çıktıları, verilerin daha sık güncellenmesi veya yazılması nedeniyle daha yüksek bir miss oranı göstermektedir.

Sonuç:

Teorik hesaplamalar, başlangıçta yüksek miss oranları ve düşük hit oranları öngörmektedir. Ancak valgrind çıktıları, özellikle komut erişimleri için çok yüksek hit oranları göstermektedir. Bu fark, gerçek dünya senaryolarında önbelleğin kullanım şekli, veri ve komut erişimlerinin sıklığı ve düzeni gibi faktörlerden kaynaklanmaktadır. Valgrind çıktıları, önbellek performansını gerçek çalışma koşullarında daha doğru bir şekilde yansıtmaktadır.

Oluşan süreler ile cache hit miss oranlarını karşılaştırması

Cache hit ve miss oranları, bir programın çalışma süresini doğrudan etkileyen önemli faktörlerdir. Önbellek erişimleri hızlı olduğundan, yüksek bir hit oranı programın daha hızlı çalışmasını sağlar. Diğer yandan, yüksek bir miss oranı, ana belleğe daha sık erişim gerektirdiğinden, programın çalışma süresini uzatır. Valgrind çıktılarındaki süreleri ve cache hit/miss oranlarını karşılaştırarak bu durumu analiz edebiliriz.

64 Byte Değeri için:

Valgrind çıktıları:

- Süre: ~0.115 saniye
- Komut İstatistikleri:
 - I refs: 12,934,856
 - I1 misses: 1,477
 - I1 miss rate: %0.01
- Veri İstatistikleri:

- D refs: 5,058,320
- D1 misses: 52,715
- D1 miss rate: %1.0

256 Byte Değeri için:

Valgrind çıktıları:

- Süre:* ~7.5 saniye
- Komut İstatistikleri:
 - I refs: 783,306,038
 - I1 misses: 1,445
 - I1 miss rate: %0.00
- Veri İstatistikleri:
 - D refs: 306,631,868
 - D1 misses: 16,894,756
 - D1 miss rate: %5.5

Karşılaştırma ve Değerlendirme:

1. 64 Byte Değeri için:

- Hit Oranları: Komutlar için %99.99, veriler için %99
- Süre: ~0.115 saniye
- Analiz: Yüksek hit oranları ve düşük miss oranları, programın hızlı çalışmasını sağlar. 64 byte değeri için düşük miss oranları ve yüksek hit oranları, sürenin çok kısa olmasına katkıda bulunmuştur.

2. 256 Byte Değeri için:

- Hit Oranları: Komutlar için %100, veriler için %94.5
- Süre: ~7.5 saniye
- Analiz: Komutlar için hit oranı %100 olsa da, veriler için miss oranı %5.5'tir. Bu, veri erişimlerinin daha sık ana bellekten yapılmasını gerektirir ve programın çalışma süresini önemli ölçüde uzatır. 256byte değeri için daha yüksek miss oranları, sürenin uzun olmasına neden olmuştur.

Genel Değerlendirme:

Cache hit oranlarının yüksek olması, programın daha hızlı çalışmasına büyük katkı sağlar. 64 byte değeri için düşük miss oranları ve yüksek hit oranları, programın süresini oldukça kısa tutmuştur. Diğer yandan, 256 byte değeri için veri miss oranlarının yüksek olması, sürenin önemli ölçüde uzamasına neden olmuştur. Bu sonuçlar, cache hit ve miss oranlarının program performansı üzerindeki etkisini açıkça göstermektedir.

Bu analiz, önbellek optimizasyonunun, özellikle veri erişimlerinin düzenlenmesinin, program performansını iyileştirmede ne kadar önemli olduğunu vurgular. Düşük miss oranları ve yüksek hit oranları, önbellek performansının artırılmasına ve programın daha hızlı çalışmasına katkıda bulunur.