

# מערכות הפעלה - תרגיל בית 4

אוניברסיטת חיפה

סמסטר אביב תשפ"ד

לתרגיל זה שני חלקים - תיאורטי ומעשי.

1. חלק תיאורטי (10 נקודות), בו יהיה עליכם לענות על מספר שאלות תיאורטיות.
2. חלק מעשי (90 נקודות), בו יהיה עליכם לממש תוכנית המדמה תרגום כתובות וירטואליות לפיזיות.

## חלק תיאורטי

1. ענו נכון/לא נכון עבור כל אחת מהטענות הבאות (נמקו).
  - (א) במידה וקיים גודל אחיד של מסגרות, בוודאות אין פרגמנטציה פנימית.
  - (ב) במידה ומשתמשים בטבלת דפים היררכית, לא נחסוך במקום אם נעבור לטבלה לינארית.
2. נתונה מערכת עם מרחב כתובות של 64 ביט, דפים בגודל 8KB, דיסק בנפח 32GB, טבלת הדפים היא לינארית ומוקצית תמיד על ה-RAM, ומכילה בכל כניסה מצביע לדף בדיסק ו-4 ביטי בקרה. מהו גודל הזיכרון הנדרש לאחסון טבלת הדפים עבור תהליך בודד?

## חלק מעשי

**הערה.** קראו את הנחיות התרגיל תוך כדי סקירה של קוד השלד המצורף - יש התייחסויות לשמות משתנים הנמצאים בקוד.

⊕ בתרגיל זה תממשו תוכנית המדמה את תהליך התרגום של כתובת וירטואלית לכתובת פיזית, ע"י טבלת מיפוי לינארית. נניח שגודל הטבלה הוא 256 דפים, וכי גודל כל דף הוא 256 בתים.

⊕ בנוסף, תממשו מנגנון TLB פשוט, שישמור תרגומים שנעשו בעבר. הניחו כי טבלת ה-TLB תקפה לאורך כל ריצת התוכנית.

⊗ שימו לב - טבלת ה-TLB הינה בגודל סופי, והכתיבה אליה תתבצע באופן ציקלי (לאחר הכתיבה לתא האחרון, בכתיבה הבאה נדרוס את התא הראשון).

⊕ התוכנית מקבלת כקלט שני קבצים:

⊗ `BACKING_STORE.bin`: זהו קובץ בינארי שנרצה לטעון לזיכרון הפיזי המדומה (לשם התרגיל), `main_memory`, ולקרוא ממנו ערכים.

⊗ `addresses.txt`: קובץ טקסט המכיל כתובות וירטואליות לתרגום.

⊕ על המשתמש שמריץ את התוכנית לספק את שני הקבצים (כמתואר בשלד): על התוכנית לקרוא כתובות וירטואליות בזו אחר זו מקובץ הכתובות. עבור כל כתובת `logical_address`, עליה לבצע את הפעולות הבאות:

1. קדם את `total_addresses` ב-1 (מונה את מספר הכתובות שנדרשנו למפות).

2. חשב את `logical_page` באמצעות `logical_address`.

3. בדוק האם `logical_page` נמצא ב-TLB.

במידה וכן, יש להשתמש ב-`physical_page` הנמצא ב-TLB, ולקדם את `tlb_hits` ב-1.

4. אם הדף הלוגי לא נמצא ב-TLB:

(א) בדוק האם `logical_page` ממופה ע"י טבלת הדפים `pagetable`. במידה

ואינו ממופה, ספגנו `page-fault exception`, ונבצע את הפעולות הבאות:

- i. קדם את `page_faults` ב-1.
  - ii. הגדר את `free page` להיות ה-`physical_page` שמתאים ל-`logical_page`.
  - iii. קדם את `free page` ב-1 (מקצים את הדפים בסדר עולה, החל מ-0).
  - iv. העתק את מקטע הזיכרון שמתאים ל-`logical_page` ב-`BACKING_STORE` למקטע הזיכרון שמתאים ל-`physical_page` ב-`main_memory`.
  - v. עדכן את ה-`pagetable` כך ש-`logical_page` ימופה ל-`physical_page`.
- (ב) קדם את `tlbindex` ב-1, ודרוס את התא במיקום `tlbindex` (הכל מודולו `TLB_SIZE`) עם הצמד `logical_page`, `physical_page`.
5. העתק את הבית שמתאים לכתובת הפיזית המלאה מ-`main_memory` ל-`value`.

## קומפילציה, הרצה ובדיקה

סופקו לכם הקבצים הבאים:

virtmem\_skeleton.c .1

BACKING\_STORE.bin .2

addresses.txt .3

output.txt .4

הדרו את התוכנית באמצעות הפקודה

```
gcc -Werror -std=c99 virtmem.c -o virtmem
```

והשוו את הפלט שלכם ל-output.txt.

## הגשה

הגשה במודל, לפי הפורמט הבא:

1. עליכם ליצור קובץ zip (השתמשו ב-zip או gzip בלבד) בשם hw4\_id1\_id2.zip, כאשר id1, id2 הם מספרי תעודות הזהות של המגישים.

2. קובץ ה-zip מכיל אך ורק את הקבצים הבאים, ללא תתי-ספריות.

⊕ virtmem.c

⊕ dry.pdf, שמכיל את התשובות לחלק התיאורטי.

⊕ submitters.txt, שמכיל את מספרי הזהות והשמות של מגישי התרגיל,

מופרדים ע"י פסיק. למשל:

Bill Gates,bill@microsoft.com,123456789
Linus Torvalds,linus@gmail.com,234567890

3. צרו את קובץ ה-`zip` באמצעות הפקודה

```
zip hw4_id1_id2.zip virtmem.c dry.pdf submitters.txt
```

4. הגישו את ה-`zip` דרך המודל.