

## החלק התאורטי: תרגיל בית 4:

### 1. א. הטענה לא נכונה.

פרגמנטציה פנימית לפי ההגדרה, הוא "חור" של זיכרון שמצד אחד מערכת ההפעלה לא יכולה להקצות אותו, אבל מצד שני אינו משומש. ולכן נוצר זיכרון שהוא מבוזבז. פרגמנטציה פנימית קורת למשל בהקצאה של מקום בגודל קבוע, שהוא גדול מדי- עבור תהליך שאינו זקוק לכל כך הרבה זיכרון.

נסמן את גודל המסגרת = גודל הדף =  $N$  גודל קבוע.

בהינתן תהליך שזקוק לזיכרון בגודל  $n < N$ , כך שבכל ריצת התוכנית מתקיים כי המקום שהוא מנצל קטן מאוד ואינו מגיע ל- $N$ . במהלך הקצאת הזיכרון עבורו: הגודל הקטן ביותר שניתן להקצות עבור תהליך היינו PAGE אחד, ולכן נקצה עבורו מסגרת בזיכרון הפיזי בגודל  $N$  שבה כל הזיכרון של התהליך. אבל נשים לב כי בזיכרון שלנו נקבל "חור" – כי מתוך המסגרת שהוקצתה עבורו בזיכרון הפיזי בגודל  $N$ - הוא משתמש בפועל רק בח מתוכה- דבר שגורם ל"חור". ה"חור" הזה הוא במרחב הזיכרון הפנימי של התהליך- התהליך אינו משתמש בו, אבל מערכת ההפעלה אינה יכולה להקצותו- כי הוא "תפוס". לפי ההגדרה, ה"חור" הזה הוא פרגמנטציה פנימית. ולכן הראינו מצב בו מתרחשת פרגמנטציה פנימית גם כאשר קיים גודל אחיד של מסגרות.

### 1. ב. הטענה אינה נכונה.

נראה דוגמא נגדית, עבור תהליך שהמקום שמוקצה עבורו בזיכרון הפיזי הוא גדול יותר כאשר הוא בטבלה היררכית מאשר אם הוא בטבלה לינארית.

כדי להראות דוגמא נגדית, מספיק להראות שישנו תהליך וישנה מערכת שמממשת טבלה היררכית או טבלה לינארית ולהראות שמתבצע חסכון בזיכרון הפיזי עבור התהליך.

הסבר אינטואיטיבי: עבור תהליך אשר משתמש ומקצה את כל הדפים שלו, והם כולם ממופים למסגרות: מתקיים כי כל הטבלה הלינארית מנוצלת ואין בה כניסות ריקות. ולכן עבור טבלה לינארית המקום הדרוש בשבילו הוא: כל המסגרות + טבלה לינארית אחת- כלומר  $N$  כניסות.

מכיוון שכל בלוק הוא מספר קבוע של כניסות בטבלת דפים לינארית, נקבל כי עבור טבלת דפים היררכית: מתקיים כי נמלא את כל הכניסות בכל הבלוקים (כי כמות הכניסות סה"כ בכל הבלוקים של תהליך אם כל הבלוקים מוקצים שווה למספר הכניסות שהיה אילו היה משתמש בטבלת דפים לינארית). מכיוון שהשתמשנו בכל הכניסה מתקיים כי בכל בלוק של התהליך – כל הכניסות מנוצלות- מה שאומר שקיימת לפחות כניסה אחת בכל בלוק שהיא מייצגת דף שמוקצה במסגרת על הזיכרון הפיזי (שזאת דרך אחרת להגיד "הכניסה מנוצלת"), ולכן מתקיים כי כל הבלוקים של התהליך גם הם מוקצים בזיכרון הפיזי. בנוסף בטבלת דפים היררכית ישנו גם מערך עם מספר כניסות ששווה למספר הבלוקים. ולכן עבור טבלת דפים היררכית מתקיים כי מוקצה על הזיכרון הפיזי:  $N$  כניסות (שמרכיבות את כל הבלוקים המוקצים) + טבלה ברמה העליונה (זו שמצינת עבור כל בלוק האם הוא מוקצה) +  $N$  מסגרות על הזיכרון הפיזי. נוכל לראות כי במקרה הזה הזיכרון של הטבלה ההיררכית גדול יותר כי הוא מכיל את כל הזיכרון שדרוש עבור טבלה לינארית + טבלה נוספת.

ולכן עבור מעבר לטבלה לינארית: נחסוך במקום.

מה שתואר כאן הוא הרעיון... לשם הפשטות פשוט נראה מצב כזה עבור אינטל 32.

(בחישובים לא הזכרנו את TLB: בכל פעם שאמרנו מה המקום הדרוש, הכוונה היא ללא קשר לTLB).

תהליך שמנצל את כל 1M הדפים שהוא יכול להקצות, כלומר ישנן 1M מסגרות בזיכרון הפיזי שמוקצות עבורו. בטבלת דפים לינארית: ישנה הטבלה הלינארית עם 1M כניסות: גודל כל כניסה הוא 4 בתים- סה"כ הזיכרון הדרוש הוא 4MB (עבור הטבלה) + 1M מסגרות.

עבור טבלת דפים היררכית: אמרנו שהתהליך מקצה את כל 1M הדפים שלו למסגרות, ולכן כל 1M הכניסות שסה"כ בכל הבלוקים שלו ביחד מנוצלות- ולכן כל הבלוקים מוקצים (בלוק מוקצה אם לפחות אחת הכניסות בו מנוצלת). כל הבלוקים ביחד המקום בשבילם היינו בדיוק במקום הדרוש עבור טבלת דפים לינארית: 1M כניסות סה"כ, כל כניסה 4 בתים: סה"כ 4MB. בנוסף ישנה את טבלת הבלוקים PGD: שאומרת לכל בלוק האם הוא מוקצה או לא : בטבלה זו 1024 כניסות (כמספר הבלוקים). ולכן סה"כ מקום הדרוש בשימוש בטבלת דפים היררכית היינו: 1M של מסגרות + 4MB עבור סה"כ הבלוקים ביחד + מערך עם 1024 כניסות (טבלת הבלוקים)- כאשר בכל כניסה 4 בתים כדי לשמור כתובת פיזית של הבלוק: ולכן מתקיים כי טבלת הבלוקים בדיוק נכנסת בPAGE אחד- אז מקצים עוד מבגרת בשבילה. ונקבל כי מספר המסגרות שמוקצות עבור התהליך בטבלת דפים היררכית, גדול ב1 ממספר המסגרות שמוקצות לו בטבלה לינארית- ולכן אם נעבור לשיטה זו נחסוך במקום.

2. נחשב את offset בכתובת הוירטואלית.  $frame\_size = 2^{|offset|}$  . ולכן

$$offset = \log_2(frame\_size) = \log_2(8KB) = \log_2(2^{13}B) = 13bits$$

הכתובת הוירטואלית היא בגודל 64 ביט: והיא מורכבת מביטים עבור offset וביטים

עבור page\_number . ולכן מספר הביטים עבור page\_number היינו

$$64 - 13 = 51 bits . ולכן עבור התהליך יש  $2^{51}$  דפים בזיכרון הוירטואלי שלו.$$

כלומר בטבלה הלינארית יש  $2^{51}$  כניסות. נחשב מה גודל כל כניסה.

נחשב את כמות המסגרות שקיימות בדיסק (כלומר כמות המסגרות שעליהן התהליכים יכולים להקצות דפים).

$$\#frames = \frac{disk\_size}{frame\_size} = \frac{disk\_size}{page\_size} = \frac{32GB}{8KB} = \frac{4 \cdot 2^{30}B}{2^{10}B} = 4 \cdot 2^{20} = 2^{22}$$

ישנן  $2^{22}$  מסגרות ולכן צריך 22 ביטים כדי לאנקס frame\_number.

כל כניסה בטבלה לינארית מורכבת מביטי בקרה, ומframe\_number: ולכן גודל כל כניסה הוא 26 bits = 22 + 4. מכיוון שצריך להשתמש בחזקות של 2, מתקיים כי כל כניסה היא הגודל 32 ביטים: כלומר 4 בתים.

$$4B \cdot 2^{51} = 8B \cdot 2^{50} = 8PB$$
 ולכן גודל הטבלה הלינארית היינו: