

## פתרון מוצע לבחינת מה"ט במערכות הפעלה

מועד א' תשע"ט, יולי 2019

מחבר: חיים ויינגולד, מכללת אורט רחובות.

### חלק א

#### שאלה מס' 1

##### סעיף א :

- **מנגנון תקשורת מאובטחת** - המערכת חייבת להכיל מנגנון תקשורת דרכו יתבצע הקשר בין מחשבי השחקנים למחשב במשחק.
- **ניהול ותזמון תהליכים ותורים** – מנגנון זה יקצה תור לניהול תהליכי השחקנים והפניות למשאבים השונים.
- **ניהול מעבד** – לספק שירות יעיל והוגן לכל המשתמשים חשוב כי המתזמן ידאג לחלק בתבונה את זמן המעבד בין המשתמשים השונים .
- **מנגנון זיהוי משתמשים** - מנגנון זה הכרחי לזיהוי השחקנים בבואם להתחבר למחשב המשחק.
- **אבטחה והצפנה** – במערכת שמורים נתוני השחקנים כולל פרטי כרטיסי האשראי לכן קיימת חשיבות רבה למנגנון אבטחת מידע כדי למנוע גניבה או שימוש לא חוקי במידע השמור במערכת.
- **מנגנונים לאיתור וטיפול בבעיות** – כל משתמש מפעיל תהליך ולעתים מפעילים כמה בבת אחת, עלול להיווצר מצב כגון הרעבה לדוגמה, אם אדם לא מקבל את ההודעה שמאשרת שנרשם למערכת או על אף שהגיע תורו לא נכנס למשחק. כדי לשמור על רמת שירות הוגנת יש חשיבות רבה לאיתור תקלות troubleshooting
- **בקרת מניעה הדדית וטיפול ב בו זמניות** – חייבים למנוע מתהליכים לעדכן את אותם נתונים בו זמנית ע"מ לשמור על שלמות הנתונים ונכונותם.
- **בסיס נתונים מקומי** – ע"מ לשמור על פרטי השחקנים ועל תוצאות המשחקים וסכומי הכסף שהושקעו עד עתה .

##### סעיף ב :

- **מנגנון הדפסות** – לא נדרש SPOOLER ותיקיית SPOOL כי לא מוזכר בשאלה הדפסות.
- **מנגנון דואר** – פחות דומיננטי למערכת.
- **מנגנון וירטואליזציה** – לא חיוני במערכת זו, אין צורך לדמות את השרת המחשבי השחקנים.

## שאלה מס' 2

### סעיף א :

משתמשים מקושרים לשרת אחד :

### יתרונות –

- **חיסכון בעלויות** – שרת אחד שעליו בסיס נתונים יחיד .
- **קלה לתחזוקה** – יש צורך בתחזוקה של שרת יחיד.
- **קלה לאבטחה** – קל יותר לבצע מנגנון אבטחה יחיד במערכת .
- **אמינות גבוהה** – בסיס בנתונים בשרת מתעדכן בזמן אמת כל הזמן ויש פחות סיכוי לתהליכים שיתבצעו עם שגיאות על נתונים לא מעודכנים, דבר המציג תמונת DB ארגונית מלאה ועדכנית כל הזמן.

### חסרונות –

- **זמני תגובה איטיים** – בגלל שמדובר על ריבוי משתמשים המנסים לפנות את מאגר הנתונים נוצר "צוואר בקבוק" בגישה למאגר הנתונים דבר המאט את זמני התגובה .
- **שרידות נמוכה של המערכת** – עקב העובדה שקיים שרת אחד , פגיעה בו משתקת כליל את עבודת המערכת.

### סעיף ב :

משתמשים מקושרים לשלושה שרתים:

### יתרונות –

- **זמני תגובה מהירים** – העומס יתחלק בין שלשה שרתים, דבר שיקצר את זמני התגובה למשתמשים .
- **שרידות מערכת גבוהה** – אם שרת אחד נפגע שאר השרתים יכולים להמשיך ולתפקד.

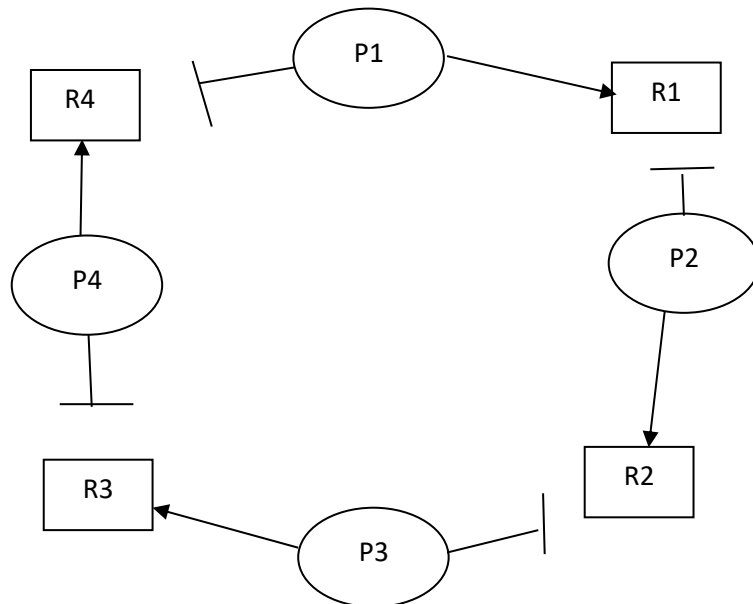
### חסרונות –

- **עלות יקרה** – מאם שקיימים שלושה שרתים שונים המשמעות רכישה ואחזקה של שלושה מחשבים, דבר המעלה את העלויות.
- **אבטחה קשה** – יש לבנות שלוש מערכות אבטחה נפרדות , אשר לא בהכרח רמת האבטחה יכולן תהיה אחידה כל הזמן , חדירה לא רצויה לאחד השרתים בעל רמת אבטחה נמוכה חושפת את כלל המערכת לפגיעה.
- **הסתמכות על סנכרון בין השרתים** – השרתים צריכים לשמור על סנכרון קבוע ע"מ שהנתונים יהיו מעודכנים ביניהם , פגיעה מסיבה כלשהי בפעולת הסנכרון תגרום לכך שהעבודות עלולות להתבצע על נתונים לא נכונים.

## חלק ב

### שאלה מס' 3

א. במצב זה עלול להתקיים מצב של DEADLOCK, במקרה שכל תהליך רוכש את המשאב מימינו וימתין למשאב שמשמאלו ( בדומה לבעיית הפילוסופים ) תתקבל המתנה מעגלית. להלן תיאור המצב שבו נקבל DEADLOCK :



מקרא :

← רכישת משאב קריטי

⊥ חסימה בעקבות משאב קריטי תפוס

ב. התנאי שיש להוסיף הוא : תהליך לא יכול לבקש משאב מסדר נמוך, כאשר הוא מחזיק משאב מסדר גבוה, על התהליך לשחרר תחילה את המשאב מסדר בבואו בלנסות לרכוש את המשאב מסדר נמוך. ( העיקרון שנלמד באלגוריתם constrained allocation שנלמד כאחת מהשיטות למניעת deadlock ) הסבר נוסף: כל תהליך הזקוק לשני משאבים מנסה לתפוס את שני המשאבים, אם התהליך הצליח הוא ממשיך להחזיק בהם עד לסיום המשימה אם הצליח לתפוס רק משאב אחד הוא משחרר אותו עד לסבב הבא. ניתן להשתמש באלגוריתם Two-Phase Locking נעילה בשני מעברים)

## שאלה מס' 4

א. שלוש סיבות הגורמות לקריסת המחשב ושיתוקו שלא ע"י הפסקת חשמל ובעיות חומרה הן:

1. מצב של DEADLOCK .
2. הרעבה .
3. לולאה אין סופית .
4. קבצים פגומים.
5. מערכת הפעלה לא מאובטחת.
6. בעיית חומרה

ב. מנגנון לטיפול בכל גורם :

1. טיפול בבעיית DEADLOCK – בבעיית DEADLOCK ניתן לטפל בשתי גישות :

א. מניעת DEADLOCK , תהליך מניעת DEADLOCK מבטל את האפשרות להמתנה במערכת וניתן למימוש בשני אלגוריתמים שונים :

אלגוריתם constrained allocation - לפי אלגוריתם זה לכל משאב קריטי יש מספר סידורי. תהליך לא יוכל לנסות לרכוש משאב שמספרו הסידורי נמוך ממספרו הסידורי של משאב שהוא מחזיק בו.

אלגוריתם הבנקאי –

לפי אלגוריתם זה לכל תהליך ידוע מראש מה הם המשאבים הקריטיים שהוא יכול להשתמש בהם כאשר תהליך פונה לראשונה למשאב קריטי בודקת המערכת האם כל הקטעים הקריטיים שהוא אמור להשתמש פנויים רק במצב זה "מצב בטוח" מאפשרת המערכת לתהליך לקבל את המשאב ונעלת את כל המשאבים שהוא יכול להשתמש. ( ביצוע הלוואה של משאב קריטי רק אם בטוחים שהתהליך יוכל להחזיר אותו ).

ב. גילוי מצב של DEADLOCK והיחלצות :

עבור כל תהליך הנכנס להמתנה מתבצעת בדיקה מה גרם להמתנה ( אם תהליך חסום בגלל שתהליך התופס את המשאב חסום גם הוא ממשיכה בדיקה על התהליך שתופס את המשאב וחסום ) אם הבדיקה תוביל אל התהליך החסום בחזרה – אזי קיבלנו Circular wait ויש להפעיל מנגנון היחלצות מקיפאון. תהליך ההיחלצות נעשה בד"כ ע"י הריגת התהליך.

2. טיפול הרעבה- בבעיית ההרעבה ניתן לטפל ע"י מימוש מנגנון AGEING . מנגנון זה מגדיל באופן אוטומטי כל פרק זמן מסוים הקבוע מראש במערכת את רמת העדיפות של התהליך דבר שיאפשר לו בסופו של דבר להיות בראש תור ה- READY ולזכות ב-CPU

3. טיפול בתהליך החשוד בלולאה אין סופית -

לכל תהליך הנוצר בשלב ה- NEW יוגדר ערך "זמן ריצה מקסימאלי" וירשם ה- PCB. כאשר תהליך מסיים CPU BURST ועובר ל- READY יוחסר מערך זה 1 . תהליך שערך "זמן הריצה המקסימאלי" שלו הגיע לאפס, מוגדר כתהליך חשוד שנמצא בלולאה אין סופית . במקרה שהתהליך החשוד הינו תהליך של USER תופיע הודעה למסך המתריעה ל- USER שהתהליך לא מגיב ותאפשר לו להמתין או להרוג את התהליך. במקרה שהתהליך הינו תהליך מערכת ההפעלה, המערכת תקבע מה גורלו.

4. קבצים פגומים:

מנגנון לטיפול וסריקת קבצים , שיבדוק את הקובץ לפני הרצה

5. מערכת הפעלה לא מאובטחת:

מנגנון אבטחה שיכלול אנטי וירוס , לדאוג לעדכוני אבטחה של מערכת ההפעלה והאפליקציות .

## שאלה מס' 5

א. בעיית הפניות המרובות לדיסק הקשיח עלולה לנבוע משלושה גורמים :

1. הקצאת מסגרות לא מספקת עבור התהליכים .

2. שימוש באלגוריתם דפדוף לא יעיל.

3. קביעת גודל דף לא מתאים במערכת .

ב. הפתרון לכל בעיה שהוצגה בסעיף א :

1. **פתרון לבעיית הקצאת מסגרות לא מספקת עבור התהליכים :**

הפתרון לבעיה זו תלוי בשיטת הקצאת המסגרות לתהליך , קיימות שתי שיטות הקצאה עיקריות **לוקאלית וגלובלית** ,

**הקצאה לוקאלית** – בגישה זו מקצה מערכת ההפעלה מראש לכל תהליך את מספר המסגרות שלו, מסגרות אלה ברשותו כל עוד התהליך רץ ולאף תהליך אחר לא ניתנת הגישה למסגרות אלו.

בשיטת הקצאה זו רצוי להקצות לכל תהליך מס' מסגרות באופן יחסי לכמות הדפים הכוללת לתהליך.

**הקצאה גלובלית** – בגישה זו לא מוקצה לכל תהליך מס' מסגרות קבוע וכמות המסגרות המוקצות לתהליך משתנה בהתאם לבקשות התהליך.

בשיטה זו רצוי להגדיר ולמצוא לכל תהליך מה הדפים החשובים והכי שימושיים עבורו ולדאוג שהתהליך מקבל את המסגרות עבור דפים אלה לכל זמן ריצת התהליך (WORKING SET).

2. **פתרון לבעיית שימוש באלגוריתם דפדוף לא יעיל:**

מטרת אלגוריתם הדפדוף הינה לבחור את הדף הנכון להוצאה מתוך כוונה לביצוע מינימום דפדופים, לאור העובדה שלא ניתן לחזות איזה דף יצטרך התהליך בעתיד ומתי, יש לנסות לקבל ניחוש מושכל על סמך העבר.

אלגוריתמים שמבצעים זאת בצורה יעילה הינם : LRU או 2nd chance.

3. **פתרון לבעיית קביעת גודל דף לא מתאים במערכת :**

גודל הדף צריך להתחשב בבעיות בזבז זיכרון , חורים פנימיים בדף וכן גודל טבלאות הדפים, שיקולי בזבז הזיכרון מנוגדים זה לזה , ככל שהדף גדול מקטינים את הבזבז של טבלאות הדפים אך מגדילים את הבזבז של החורים הפנימיים ולהפך. גודל הדף גם צריך לקחת בחשבון את גודל התהליך הממוצע וכן את גודל כניסה בטבלת הדפים .

גודל דף אידיאלי הינו :  $\sqrt{2 * S * E}$

כאשר : S – גודל תהליך ממוצע

E – גודל כניסה בטבלת הדפים

## שאלה מס' 6

א. בעיבוד עבודה אחת בלבד, העבודה זוכה לכל משאבי המחשב, גם אלו שאינה זקוקה להם, משאבים רבים יעמדו במצב – idle חוסר מעש. בזמן פסיקה שום עבודה אחרת לא תיכנס להרצה, המעבד "לא יעבוד", עד החזרה של התוכנית מהפסיקה. היתרונות בשיטה הוא שאין תהליכים שמחכים לשחרור התקן, אין תורים, דד לוקים ואין הרעבה ועל כן זמן סבב ממוצע יהיה נמוך יותר. הפנייה להתקן תהיה נטולת בדיקות ונטולת תקורה הכרוכה בזימון תהליכים, לכן ההיענות של המערכת להקצאת משאב תהיה מידית (=קיצור זמן התגובה של כל תהליך).

ב. ביצוע של יותר מעבודה בו זמנית מייעל את המחשב כי:

- (1) עבודה מסוימת מנצלת משאב אחד בזמן נתון ואילו עבודה אחרת, הזקוקה למשאב אחר באותו הזמן, תוכל להשתמש בו אם הוא פנוי לעבודה ולכן נצילות הצידוד ההיקפי גבוהה.
  - (2) אם קיימים משאבים זחים ניתן להקצות אותם במקביל ליותר מעבודה אחת (לדוגמה שתי מדפסות). לפיכך מושגת נצילות גבוהה יותר של הצידוד ההיקפי.
  - (3) ניצול המשאבים עולה וכך גם התפוקה הכללית של העבודות המבוצעות. הן יבוצעו יותר מהר, כי הן מבוצעות במקביל.
- ביצוע של יותר מעבודה אחת בו זמנית מייעל את עבודת המחשב תלוי בכך שהיא מייעל את עבודת היישומיים של פעולה ופעולה (בתהליכונים) ע"י גוש בקרת התהליך, בכך שמאפשר קצב תזמון טוב הוא יחלק את עבודה בתור הממתנים ייקח עבודה יעשה ימין ינסה לעשות בדיקה בעזרת שיבדוק מה חשוב ומתי צריך אבל יכול להיווצר בעיה שהמעבד יהיה תפוס מפני שפעולה אחת נתקעת ולא יהיה לה פתרון אז במקרה כזה דבר לא יעיל לעבוד יותר מכמה תהליכים.
- לסיכום עדיף שעבוד בו זמני כי זה מייעל את עבודה והתזמון בין תהליכים שלא ייווצר מצב קריטי, המעבד עובד בעזרת תור הממתנים לוקח תהליך ויש לו תור מושעים שיכול להשים תהליכים שממתנים לתשובה.

שני המנגנונים החשובים שמאפשרים ביצוע של יותר מעבודה אחת בו זמנית:

- (1) מנגנון ניהול תורים במתזמן – על מנת שתהליכים יחכו כמה שפחות להתקן פופולרי כלשהו. על מנת למנוע גישה בו זמנית למשאבים. כדי שכל התהליכים יקבלו הזדמנות הוגנת ולא ימתינו זמן רב מדי לתחילת הרצה. בכך שהתהליכים ימתנו בתור הממתנים עד שהמעבד ייקח משימה אחת יבצע אותה ויחזיר אותה לתור או לתור המושעים. מנגנון התזמון עוזר למעבד במתן זמינות למשימה ומשימה, וגם כן עוזב מתן קצביים וסדר איך התהליכונים יסדרו בתור הממתנים ואיזה תהליכונים סיימו.

- (2) מנגנון לניהול משאבים – על מנת שיהיה ניתן לחסוך זמן (של המתנה למשאב) וכסף (בקניית משאבים זחים) ולהקצות משאב לכמה תוכניות במקביל. מנגנון התהליכים עוזר לסדר את התהליכונים למשימות ומפרק כול תהליך לתת-משימות, וגם מסדר לפי חשיבות הדחיפות יישאר בתור הממתנים בעזרת מנגנון התזמון, וגם - כן עוזר להעביר בסוף לזיכרון הראשי תהליכים שסיימו עם "צילום המצב".

ניתן להפעיל על מחשב זה מערכת הפעלה בשתי גישות פעולה שונות :

1. מערכת עיבוד מקבילית סימטרית - Symmetric Multiprocessing שבה על כל מעבד רצה אותה מערכת הפעלה. כל מערכות ההפעלה האלה מקושרות ביניהן ע"י פס תקשורת משותף הנקרא bus המבטיח כי כל קלט יגיע למעבד הנכון. במקרה זה מספר תהליכים יכולים לרוץ במקביל ובו זמנית בלי לפגוע בביצועים.

#### יתרונות:

- אמינות גבוהה ואפשרות המשך תפקוד חלק של המערכת במקרה של נפילת מעבד.
- חלוקה טובה של עומס העבודה בין המעבדים, כיוון שכל משימה יכולה להישלח למעבד אחר.
- בכל רגע נתון יש רק מעבד "אדון" אחד בלבד, תופעה שמונעת התנגשויות בין מעבדים.
- מעבדים יכולים לשתף פעולה ביניהם לצורך ביצוע של תהליך של משתמש.
- אחוזי הנצילות של כל אחד מהמעבדים גבוהה יותר מאשר במערכות אחרות.

### חסרונות:

- הקוד של מערכת ההפעלה חייב להיות שיתופי ודרושה בלעדיות.
- בעיות של תחרות בין המעבדים יכולה לגרום להתנגשויות חומרה ותוכנה.
- נדרש מנגנון מיוחד נגד התנגשויות הן בחומרה והן בתוכנה.
- עלולות להיווצר חסימות במערכת בגלל תנועה רבה בין המעבדים.
- נדרשת תקורה נוספת עבור מערכת ההפעלה,
- יש תחרות רבה יותר ובמקביל על משאבי המערכת, חומרה ותוכנה.
- מתרחשים עיכובים בגלל מעבר מידע ניהולי רב בין ה"אדון" לבין המעבדים "העבדים".

2. מערכת עיבוד אסימטרית – Asymmetric Multiprocessing שבה ישנו מעבד אחד ראשי המריץ את מערכת ההפעלה ותפקידו להקצות משימות למספר מעבדים משניים באופן לא סימטרי. לכל מעבד יש את המשימות שהוקצו עבורו.

### יתרונות:

- מניעת חסימות הדדיות אינה קשה יותר מאשר במערכות מקבילות אחרות.
- אין צורך בקוד שיתופי כי כל מעבד מפעיל גרסת מערכת הפעלה משלו ואין הוא זקוק לקוד שיתופי כי רק הוא מבצע את התהליך בזמן נתון.
- מתאים למערכות בהן יש עיבוד רב מאוד וצורך רב ביכולות עיבוד עתיר תהליכי חישוב.
- מתאים למערכת בעלת כושר עבודה יציב וקבוע ללא שינויים תכופים.
- אופטימלי למערכת בעלת מתזמן פשוט ולא מורכב.

### חסרונות:

- "העבדים" יכולים להריץ רק תהליכים עתירי חישוב.
- כל תהליך קלט פלט דורש לפחות שתי רמות של פסיקה.
- במערכות שהתזמון שלהן מורכב יותר יהיו תורים ארוכים אצל המעבד "האדון".
- נפילת מעבד "אדון" גורמת לקריסת המערכת כולה.

## שאלה מס' 7

- א. לא נכון** - עיבוד קובץ על מספר דיסקים יאפשר גישה בו זמנית לבלוקים של הקובץ בדיסקים השונים בתנאי שעובדים ב-raid.
- ב. לא נכון** - תהליך שנימצא ב WAIT מוותר על ה CPU מרצונו, סיבות לכך יכולות להיות: המתנה לפעולת IO, המתנה לתוצאה מתהליך אחר וכו'.
- ג. נכון** - כיוון שסרט מגנטי שומר את הנתונים בצורה רציפה, כדי להגיע לבלוק מסוים יש לעבור כל הבלוקים הקודמים לו באופן סידרתי, בעוד שבדיסק ניתן להגיע באופן ישיר לבלוק מסוים לצורך קריאתו.
- ד. נכון** - מנגנון הפסיקה גורם להפעלת שיגרת לטיפול בבקשת הפסיקה אשר כתובתה ממופה בווקטור הפסיקה.

## שאלה מס' 8

א. הבעיות העלולות לקרות במערכת בעקבות תוכניות שאינן בשימוש :

- **האטה בביצועים** – הרבה תוכניות מעמיסות על המערכת כאשר אין בהם שימוש כך שהן תופסות מקום בזיכרון ( מסגרות ) , מקום שיכול להיות בשימוש תוכניות שאכן פעילות וצריכות אותן.
- **אחזקת משאבים באופן בלעדי** – תוכניות אלה עלולות להחזיק במשאבים הדרושים לתוכניות אחרות באופן בלעדי דבר העלול לגרום ל DEADLOCK במערכת.
- **תקלות בפתיחת קבצים ובתעבורת קבצים** – שאילות על הנתונים יהיו שגויות ולא עדכניות.

ב. תפעול מנגנון בדיקת שכיחות השימוש בתוכניות כדי להתריע על תוכניות ישנות ללא שימוש והמלצה להסירן מהמחשב משולב עם מנגנון חסימות שיגרום לתוכניות שאינן בשימוש שכיח להפסיק לעדכן את מסדי הנתונים.

## שאלה מס' 9

תהליך העתקת הקובץ לקובץ עם שם אחר ומחיקת הקובץ המקורי הינו תהליך הדורש פעולות רבות של גישה לדיסק .

הפעולות הנדרשות לצורך העתקת קובץ :

- איתור inode פנוי במערך ה - inode של מע' הקבצים.
- הוספת שורה בטבלת האובייקטים בתיקיה שאליה משויך הקובץ והפניה ל- inode
- העתקת תוכן ה - inode המקורי של הקובץ ל - inode שנימצא ( קריאה וכתובה )
- איתור בלוקי data פנויים בדיסק .
- העתקת תוכן בלוקי ה - data המקוריים ל בלוקי ה data שנמצאו.
- יצירת טבלאות הצבעה ל בלוקי ה - data של הקובץ החדש ( level1, level2,level3 ).
- קישור טבלאות ההצבעה ש ל - inode של הקובץ החדש.

פעולות הדרושות למחיקת קובץ :

- שחרור כל בלוקי ה - data של הקובץ וכן בלוקי ההצבעה ( טבלאות המצביעים ) והפיכתם לבלוקים פנויים. ( הוספתם לרשימת הבלוקים הפנויים ב super block בדיסק )
- שחרור ה- inode של הקובץ והפיכתו לפנוי . ( הוספה שלו לרשימת ה inodes הפנויים ב super block בדיסק ) .
- מחיקת השורה של הקובץ בטבלת האובייקטים בתיקיה שאליה היה שייך הקובץ .

תהליך שינוי שמו של קובץ פשוט ביחס לתהליך העתקת קובץ בשם אחר ומחיקת הקובץ המקורי.

לצורך שינוי שמו של קובץ נדרשת גישה אל טבלת האובייקטים של התיקיה אליה שייך הקובץ ושינוי השם בשורת הקובץ בטבלה.



## שאלה מס' 10

**הערה :** בשאלה לא מצוין כמה בלוקי data תופסת טבלת האובייקטים של התיקיה etc וכן כמה בלוקי data יש לקובץ passwd .  
הפתרון מתייחס בהנחה כי גודל טבלת האובייקטים של התיקיה etc הוא בלוק אחד ושלקובץ passwd בלוק data יחיד.

נאמר בשאלה כי ROOT BLOCK כבר בזיכרון כך שאין צורך לפנות לדיסק עבורו.

מס' הגישות לדיסק היא : 4 .

פרוט הגישות :

1. גישה ל inode של התיקיה etc .
2. גישה לטבלת האובייקטים השייכים לתיקיה etc ( etc block ) .
3. גישה ל- inode של הקובץ passwd .
4. גישה לבלוק ה data של הקובץ passwd .