

# תרגיל # 4

## חלק ראשון

### שאלה 1 NFS

א. הגדירו מהו ה- file system (מערכת קבצים)?

מערכת קבצים (File System) היא השיטה המשמשת למחשבים ולמערכות הפעלה לארגן ולאחסן נתונים על תקליטים קשיחים, דיסקים קשיחים, או כל תקליט אחסון אחר. מערכת הקבצים מספקת דרך מאורגנת לאחסון, לאחזור ולניהול של קבצים ותיקיות במערכת.

המערכת קבצים מחליטה ומספקת איך נראה disk layout :

1. איך התוכן שרוצים לשמור נשמר על הדיסק ובאיזה היררכיה.
2. איזה סוג גישה יש (open, read, write, seek, close), ומאפשרת גישה למי שמותר לו
3. ממפה בין <file,offset> ל- <sector,offset>
4. עוקבת אחרי ענפים "תפוסים" לענפים ריקים בזכרון
5. פיצ'רים נוספים - מתודות לשיפור אמינות, דחיסה, הצפנה, ניהול, שמירת מצב snapshot

הסבירו בקיצור מהם תפקידי מערכת ההפעלה (operating system) בכל הקשור לטיפול בקבצים:

מערכת ההפעלה מספקת תצוגה אחידה ולוגית של אחסון מידע:

- יצירה ומחיקה של קבצים וספריות (directories) - נותנת לתוכנות גישה לקבצים באמצעות ממשק תוכנה (API) המספק פונקציות כמו פתיחת קובץ, קריאה ממנו וכתיבה אליו.
- מתפעלת קבצים וספריות
- מיפוי קבצים לאחסון משני
- גיבוי קבצים על מדיית אחסון יציבה (לא נדיפה)
- משנה בקורות גישה - כולל הגדרת מי יכול לקרוא, לכתוב או למחוק קבצים ותיקיות מסוימות.
- ניהול מטה-נתונים (Metadata): מערכת הקבצים מאחסנת מידע נוסף על כל קובץ או תיקייה, כגון תאריכי יצירה ושינוי, גודל, מיקום פיזי על הדיסק והרשאות גישה.
- ניהול משאבים פיזיים: מערכת ההפעלה נותנת לכל תוכנה גישה למשאבים הפיזיים במחשב, וכולל גישה לתקליטים קשיחים או אחסון אחר עליו נשמרים הקבצים
- ניהול התמיכה בפורמטים שונים: מערכת הקבצים צריכה לתמוך בפורמטים שונים כגון NTFS, FAT32, או ext4 וכו' ומאפשרת קריאה וכתיבה לקבצים בהתאם לפורמט המתאים.

בקיצור, מערכת הקבצים היא התשתית הבסיסית שבה מסודרים וניהולים קבצים במחשב, ומערכת ההפעלה משמשת כגורם מנהיג שמקבע כיצד יתבצעו הפעולות השונות על הקבצים והתיקיות במערכת.

**ב. הניחו בקשת I/O הקוראת מקובץ "users/galaxy/venus/haizar/". הסבירו את פעולות ה LOOKUP הנעשות כדי להגיע ולקרא מקובץ ה haizar. כל הקבצים של התיקיות בהיררכיה זו שמורות באותה מערכת קבצים ext3FS**

1. ע"י ה mount הראשון נדע את ה inoden של הסלש הראשון. יוחזר לנו רשימה של כל התיקיות עם ה inodes שלהם. וביניהם את users יחד עם ה inode שלו. נבקש את ה inoden של users ויוחזר לנו רשימה של כל הקבצים שנמצאים שמה יחד עם ה inodes שלהם, ביניהם יהיה את ה inoden של galaxy.
2. נבקש את ה inoden של galaxy. יוחזר תשובה עם רשימה של כל הקבצים הנמצאים ב galaxy יחד עם ה inoden שלהם. ביניהם יהיה את ה inoden של venus.
3. נבקש את ה inoden של venus. יוחזר תשובה עם רשימה של כל הקבצים הנמצאים ב venus יחד עם ה inoden שלהם. ביניהם יהיה את ה inoden של haizer.
4. נחפש את ה inoden של haizer, וזה מה שיוחזר לנו. הקריאה מהקובץ באמצעות הממשק המתאים למערכת הקבצים ext3FS.

בסיום תהליך ה LOOKUP והגעה למיקום הנכון של הקובץ "haizar", מערכת ההפעלה תקבע למיקום המדויק במערכת הקבצים ותקרא את הנתונים מהקובץ כפי שנדרש בבקשת ה I/O.

הערה: מבנה ה i-node מאוחסן בדיסק ומכיל את הכתובות של בלוקי נתונים שהוקצו לקובץ.

**ג. הסבירו מה תפקיד פונקציית ה mount בהגדרה וגישה לקבצים?**

ספריית השורש (root directory) של הקבצים מורכבת מחלקים יותר קטנים הנקראים filesystem. כל filesystem מכיל directory (ספרייה), ספריות משניות וקבצים - והיא מסודרת ב file tree. רוב ה filesystem מחוברות לעץ עם הפקודה mount. הפקודה ממפה ספרייה בתוך file tree הקיים, הנקראת נקודת ההרכבה - mount point, לשורש מערכת הקבצים החדשה שצורפה. התוכן הקודם של נקודת הטעינה הופך לבלתי נגיש לאחר שמערכת קבצים חדשה נטענת שם.

בעצם, פונקציית ה "mount" במערכת ההפעלה נועדה להתחבר ולהפעיל מכשיר אחסון חיצוני כחלק מהמערכת הקיימת של הקבצים. תפקידה העיקרי של הפונקציה הוא להקצות מקום בשיטת הקבצים למכשיר החיצוני, כך שהמערכת תוכל לגשת אליו כמו לכל מכשיר אחסון פנימי.

פונקציית ה mount פועלת באופן הבא:

1. מגדירה קשר בין מערכת ההפעלה לבין מכשיר חיצוני (דיסק קשיח/כרטיס זכרון וכו') לבין היררכיה של הקבצים במערכת הפעלה.
2. מפעילה את המכשיר החיצוני ומאפשרת למערכת ההפעלה לקרוא ולכתוב אליו נתונים כאילו הם חלק מהמערכת הפעלה.
3. אפשר גם להריץ תוכניות וקבצים שמאוחסנים על המכשיר החיצוני, כאילו הם מותקנים על המחשב באופן רגיל.

**ד. הניחו בקשת I/O הקוראת מקובץ "users/galaxy/venus/haizar/". התיקיות users/galaxy נמצאות במערכת קבצים ext3FS ואילו התיקיה venus שמורה ב NFS. הסבירו את פעולות ה LOOKUP הנעשות כדי להגיע ולקרא מקובץ ה haizer**

תהליך lookup מתבצע ככה:

1. ע"י mount הראשון נדע את inoden של הסלש הראשון.  
יוחזר לנו רשימה של כל התיקיות עם inodes שלהם. וביניהם את users יחד עם ה inode שלו.  
נבקש את inoden של users ויוחזר לנו רשימה של כל הקבצים שנמצאים שמה יחד עם inodes שלהם, ביניהם יהיה את inoden של galaxy.
2. נבקש את inoden של galaxy. יוחזר תשובה עם רשימה של כל הקבצים הנמצאים בgalaxy יחד עם inoden שלהם.
3. נחפש בין כל הקבצים של galaxy את venus. כאשר מערכת ההפעלה מגיעה לתיקייה "venus", ה VFS זיהתה שהתיקייה נמצאת ב-NFS.
4. עושים mount לשרת NFS ואומרים לשרת איזה נתיב אני רוצה בתוך השרת NFS. הפעולת mount לוקח את הניתוב ומחזיר file handle אם ללקוח יש הרשאות גישה ל filesystem לקבצים שהוא רוצה. במקרה שיש לו הרשאות גישה נמשיך לשלב הבא.
5. עכשיו filehandles של שרת NFS ממופים לתוך inodes של השרת.
6. תהליך LOOKUP ב-NFS מבוצע דרך רשת, ע"י שימוש ב-RPC. תהליך LOOKUP יבקש מהשרת של NFS למצוא את התיקייה venus. נקבל את הרשימה של כל הקבצים שנמצאים שמה וביניהם יהיה את venus. נעשה חיפוש על inoden שלה.
7. ברשימה של הקבצים של venus יהיה את הקובץ haizar.
8. מערכת ההפעלה תבצע בקשה לקריאת הנתונים (מהקובץ haizar) מהשרת דרך NFS. הנתונים יועברו מהשרת אל הלקוח דרך הרשת, ע"י file handler.
9. השרת NFS עושה export (הלקוח מקבל את haizar מהפעולת mount) export אומר לנו איזה הרשאות יש לאיזה לקוח

בכל השלבים הללו, הפעולות מתבצעות תחת הנחות שהגישה לשרת NFS מוגדרת וזמינה ושהמשתמש או התהליכים שמנסים לגשת לקובץ יש להם הרשאות גישה מתאימות.

**ה. הסבירו בעזרת הפקודות read באיזה מובן פרוטוקול ה NFS הינו Stateless**  
השרת NFS הוא stateless כיון שהוא לא שומר מידע מה עושה כל לקוח. לכל בקשה נצטרך לשלוח לשרת את כל המידע הדרוש. והפעולות לא קשורות אחת לשניה, נצטרך לשלוח את כל המידע מחדש. למשל בפקודת READ נעביר גם את offset וכמה בתים לקרוא. ואז השרת יוכל לעשות את הקריאה ולהחזיר את המידע ללקוח (או שגיאה) השרת לא שומר את המישע שהעברנו לו.

#### ו. איך העובדה שה-NFS stateless-עוזרת בזמן נפילת השרת? ובזמן נפילת הלקוח?

במקרה של נפילה של שרת - כיון שהשרת לא צריך לשמור שום מידע על מה שעושה כל לקוח אז זה עוזר לנו במקרים של נפילה.  
התכונה ה- Stateless מבטיחה שאין נתונים חשובים מאוד מאוחסנים על השרת עצמו. מכיון שהשרת אינו שומר מצב או נתונים על הלקוחות, אין תלות בשמירת מידע על מצב השרת. ואז הלקוח שולח את הבקשות שלו שוב ושוב עד שמקבל מענה. (ואז בכלל אין crash recovery לשרת)

במקרה של נפילת לקוח - מכיון שהלקוח לא שומר מידע על מצב השרת או על הבקשות הקודמות שלו, השרת יכול לפספס את התקלה ולהמשיך לתפקד בצורה תקינה. כאשר לקוח חוזר לתפקוד, הוא יכול לשלוח מחדש את הבקשות אל השרת, והשרת יספק לו את התגובה המתאימה (ובעצם לא צריך שום crash recovery).

בגלל זה, התכונה ה- Stateless של NFS מקנה יכולת גבוהה לשחזור ולהתמודד עם נפילות שרת או לקוח ברשת בצורה שמשפיעה על התפקוד הכולל של המערכת באופן מינימלי.

## שאלה 2 DNS & CDN

שימוש במנגנון ה Cache בתוך המחשב ובגישה לשרתים ב WEB נועד לשפר ביצועים, זמינות למשאבים וזמני תגובה ללקוח. ה Cache יכול להיות שיתופי למספר לקוחות או פרטי ללקוח. א. פרוטוקול ומסד ה Domain Name Services (DNS) מחזיר את כתובת ה IP עבור domain נתון. תארו את המסלול של שאילתת DNS מיציאת מהדפדפן ועד קבלת כתובת ה-IP של הדומיין <https://engineering.tau.ac.il> (הפקולטה להנדסה באוניברסיטת תל אביב).

1. הדומיין נשלח את השרת local DNS. השרת מפנה אותנו לשרת ROOT DNS
2. השרת ROOT DNS לא מכיר את הדומיין הזה ולכן אומר לנו לחפש בשרת IL (ה-IL אומר ישראל)
3. שרת ה-IL מפנה אותנו לשרתים של AC, שרתי האקדמיה.
4. שרתים האקדמיה מפנים אותנו לשרת של אוניברסיטת תל אביב
5. בתוך השרת של האוניברסיטה מפנים לאתר הסופי הנדרש, של הפקולטה להנדסה בתוך אוניברסיטת תל אביב.
6. משם נקבל את ה-IP הדרוש והמתמשש יוכל לגשת את האתר.

ב. תארו כיצד שימוש ב caching יכול לשפר את תהליך קבלת המידע בשאילתת DNS cache יכול לשפר את התהליך כך -

1. המשתמש מזין את כתובת <https://engineering.tau.ac.il> בדפדפן שלו.
2. הדפדפן מפעילה http ויודעת לגשת לגוגל
3. אם ה-root נמצא בזכרון מטמון אז הרמות שמתחת גם שמה ואפשר לחסוך בפניות כי לא לעשות את כל החיפוש. וכן העומס על שרתי ה-DNS יכול להיות פחות כבד. זה מקטין את זמן התגובה ומפחית את המאמץ הנדרש מהשרתים לעיבוד שאילתות חוזרות. הפחתת העומס על השרתים עשויה להפוך את תהליך השירות למהיר יותר ולהפחית את הסיכון לעומס גדול בשרתים, שעשוי לגרום לפסקות בשירות. ואז גם יש ללקוח זמן תגובה מהיר מאוד.

## איזו בעיה עלולה להיווצר משימוש ב-DNS?

1. במקרים בהם מידע על כתובות ה-IP משתנה בתדירות (למשל, במקרה של חידוש רשומות DNS עבור אתרים חדשים או שינוי IP), המידע במטמון יכול להיות מיושן ולא עדכני. זה יכול לגרום למשתמשים להתקשות להגיע לאתרים חדשים או לקבל את התגובה הנכונה במקרים שבהם כתובת ה-IP השתנתה.
2. תקלות בזמן עדכון: אם תהליך העדכון או ניהול המטמון אינו מתבצע בצורה תקינה או תקופות קצרות מדי, זה עשוי להביא לכך שמידע מיושן יישאר במטמון לזמן רב יותר מדי. זה עשוי לגרום לתקלות בזמן עדכון ולעומס על המערכת.

בכל זאת, ברוב המקרים שימוש ב- caching ב-DNS מביא לשיפור ביצועים ולשיפור בחווית המשתמש ברשת.

ג. מאגרי תוכן גדולים משפרים את חווית המשתמש ע"י הוספת Web cache הקרויות CDN (Content Delivery Network).

מנגנון ה- DNS משתלב בתהליך ה-streaming מה CDN. תארו את הגישה לסרט <https://www.netflix.com/watch/81031849> ב-Netflix.

1. כאשר המשתמש מזין את הקישור בדפדפן שלו, ה-DNS Resolver מה-local DNS של המכשיר שלו מתרגם את שם הדומיין (www.netflix.com) לכתובת ה-IP המתאימה של השרת של Netflix.

2. השרת של netflix מחזיר את ה-URL של ה-CDN שמחזיק את הסרט הדרוש

3. ה-CDN DNS מחזיר את כתובת ה-IP של ה-CDN "טוב" (קרוב ללקוח ועם התעכבות מינימלית) או שהלקוח בוחר איזה CDN טוב לו) יחד עם הסרט (ועכשיו ב-CDN יש העתק של הסרט) ל-local DNS

4. מבקשים את הסרט מה-CDN והגישה לסרט היא באמצעות streaming מעל HTTP, ה-streaming הוא של DASH. השרת מחלק את הסרט לחלקים (chunks), וכל חלק יש לו רמת איכות משלו (יש טובות יותר ויש טובות פחות). לכל חלק יש URL משלו. הלקוח בודק כל פרק זמן איזה איכות מקסימלית הוא יכול לקבל וכך הוא יקבל את החלק הזה של הסרט.

**הסבירו איך עקרונות ה-caching באים לידי ביטוי ב-CDN ?**

עקרונות ה-Caching מושגים ב-CDN עוזרים לשפר את חווית המשתמש ב-Streaming של תוכן מסוים, כגון סרטים ב-Netflix. על ידי השמירה על עותקים של התוכן בשרתים מקומיים בקרב המשתמשים, ה-CDN מספקת גישה מהירה ויעילה לתוכן על מנת לספק חוויה חלקה וברמה גבוהה למשתמשים ברחבי העולם.