

מבנה מערכת - CPU

- * מבנה מערכת: CPU, Memory, I/O structures, OS, Interrupts, Bus, System call.
- * CPU: מבנה מערכת (CPU, Memory, I/O structures, OS, Interrupts, Bus, System call).
- * Memory: מבנה מערכת (CPU, Memory, I/O structures, OS, Interrupts, Bus, System call).
- * I/O structures: מבנה מערכת (CPU, Memory, I/O structures, OS, Interrupts, Bus, System call).
- * OS: מבנה מערכת (CPU, Memory, I/O structures, OS, Interrupts, Bus, System call).
- * Interrupts: מבנה מערכת (CPU, Memory, I/O structures, OS, Interrupts, Bus, System call).
- * Bus: מבנה מערכת (CPU, Memory, I/O structures, OS, Interrupts, Bus, System call).
- * System call: מבנה מערכת (CPU, Memory, I/O structures, OS, Interrupts, Bus, System call).

מערכת מבוזרת - Distributed System

- * מבנה מערכת: CPU, Memory, I/O structures, OS, Interrupts, Bus, System call.
- * CPU: מבנה מערכת (CPU, Memory, I/O structures, OS, Interrupts, Bus, System call).
- * Memory: מבנה מערכת (CPU, Memory, I/O structures, OS, Interrupts, Bus, System call).
- * I/O structures: מבנה מערכת (CPU, Memory, I/O structures, OS, Interrupts, Bus, System call).
- * OS: מבנה מערכת (CPU, Memory, I/O structures, OS, Interrupts, Bus, System call).
- * Interrupts: מבנה מערכת (CPU, Memory, I/O structures, OS, Interrupts, Bus, System call).
- * Bus: מבנה מערכת (CPU, Memory, I/O structures, OS, Interrupts, Bus, System call).
- * System call: מבנה מערכת (CPU, Memory, I/O structures, OS, Interrupts, Bus, System call).

מרכיבי המחשב

* Main memory - זיכרון ראשי, זיכרון מרכזי, זיכרון ראשי, זיכרון ראשי, זיכרון ראשי

* ROM (Read Only Memory) - זיכרון קריאה בלבד, זיכרון קריאה בלבד, זיכרון קריאה בלבד

* RAM (Random Access Memory) - זיכרון גישה אקראית, זיכרון גישה אקראית, זיכרון גישה אקראית

* Secondary Storage - זיכרון משני, זיכרון משני, זיכרון משני, זיכרון משני, זיכרון משני

* Hard Disk - דיסק קשיח, דיסק קשיח, דיסק קשיח, דיסק קשיח, דיסק קשיח

* SSD (Solid State Drive) - דיסק מצב מוצק, דיסק מצב מוצק, דיסק מצב מוצק, דיסק מצב מוצק

* Caching - קידוד, קידוד, קידוד, קידוד, קידוד, קידוד, קידוד, קידוד

* Dual Mode Operation - פעולה כפולה, פעולה כפולה, פעולה כפולה, פעולה כפולה

1) User Mode - מצב משתמש, מצב משתמש, מצב משתמש, מצב משתמש (mode bit = 1)

2) Monitor Mode (kernel) - מצב מונитор, מצב מונитор, מצב מונитор, מצב מונитор (mode bit = 0)

- System call - קריאה מערכת, קריאה מערכת, קריאה מערכת, קריאה מערכת

- Process - תהליך, תהליך, תהליך, תהליך, תהליך, תהליך, תהליך, תהליך

* Progress - התקדמות, התקדמות, התקדמות, התקדמות

- Thread - אשף, אשף, אשף, אשף, אשף, אשף, אשף, אשף

* Process - תהליך, תהליך, תהליך, תהליך, תהליך, תהליך, תהליך, תהליך

- Thread - אשף, אשף, אשף, אשף, אשף, אשף, אשף, אשף

- Thread - אשף, אשף, אשף, אשף, אשף, אשף, אשף, אשף

* Process - תהליך, תהליך, תהליך, תהליך, תהליך, תהליך, תהליך, תהליך

* Buffer - מחזור, מחזור, מחזור, מחזור, מחזור, מחזור, מחזור, מחזור

- Cache - זיכרון קטן, זיכרון קטן, זיכרון קטן, זיכרון קטן, זיכרון קטן

מערכת הפעלה

* מיון של משימות: * זמן תגובה, * זמן עיבוד, * זמן המתנה

- עקרונות: * זמן תגובה, * זמן עיבוד, * זמן המתנה
(הזמן הדרוש לביצוע המשימה, * זמן המתנה לביצוע המשימה, * זמן העיבוד של המשימה)
(הזמן הדרוש לביצוע המשימה, * זמן המתנה לביצוע המשימה, * זמן העיבוד של המשימה)

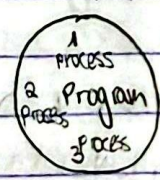
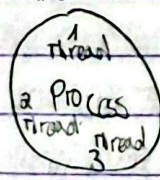
- מערכת הפעלה: * זמן תגובה, * זמן עיבוד, * זמן המתנה
* System call - מיון של משימות
מיון של משימות על ידי מערכת הפעלה (API) (מיון של משימות)

* Process - מיון של משימות
* Job - מיון של משימות (מיון של משימות)
* Task/User program - מיון של משימות (מיון של משימות)
* Program - מיון של משימות (מיון של משימות)

מיון של משימות על ידי מערכת הפעלה (API) (מיון של משימות)
מיון של משימות על ידי מערכת הפעלה (API) (מיון של משימות)
מיון של משימות על ידי מערכת הפעלה (API) (מיון של משימות)

מיון של משימות על ידי מערכת הפעלה (API) (מיון של משימות)
מיון של משימות על ידי מערכת הפעלה (API) (מיון של משימות)
מיון של משימות על ידי מערכת הפעלה (API) (מיון של משימות)

מיון של משימות על ידי מערכת הפעלה (API) (מיון של משימות)
מיון של משימות על ידי מערכת הפעלה (API) (מיון של משימות)
מיון של משימות על ידי מערכת הפעלה (API) (מיון של משימות)

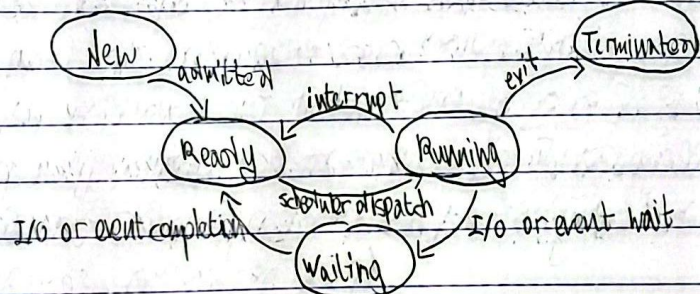


* Thread - מיון של משימות (מיון של משימות)
מיון של משימות על ידי מערכת הפעלה (API) (מיון של משימות)
מיון של משימות על ידי מערכת הפעלה (API) (מיון של משימות)

מיון של משימות על ידי מערכת הפעלה (API) (מיון של משימות)
מיון של משימות על ידי מערכת הפעלה (API) (מיון של משימות)
מיון של משימות על ידי מערכת הפעלה (API) (מיון של משימות)

מחזורי תהליך

- * process control - תהליך מנוהל על ידי מערכת ההפעלה. כולל:
 - 1. Heap (זיכרון), Stack (זיכרון), Data (נתונים), Text (טקסט) (סיפור)
 - 2. ready - תהליכים מוכנים להיבצע על ידי ה-CPU.
 - 3. running - תהליך בביצוע.
 - 4. waiting - תהליך המתחיל להמתין לסיבוב שוב.
 - 5. Termination - תהליך שהסתיים.
- * ready - תהליכים מוכנים להיבצע על ידי ה-CPU.
- * running - תהליך בביצוע.
- * waiting - תהליך המתחיל להמתין לסיבוב שוב.
- * Termination - תהליך שהסתיים.



התהליך החדש נכנס למצב ready לאחר שהתקבלה אישור מהמערכת. כאשר ה-CPU זמין, הוא יבצע את התהליך. במהלך הביצוע, התהליך יכול להיפסק על ידי interrupt או להמתין לסיבוב שוב בגלל I/O or event wait. כאשר ה-I/O או האירוע מתמלא, התהליך יחזור למצב ready. תהליכים הממתנים יחזרו למצב ready כאשר ה-CPU יבצע אותם.

* Process control block PCB - תבנית בקרה ותהליך. כוללת מידע על התהליך, כגון:

- 1. ready - תהליכים מוכנים להיבצע על ידי ה-CPU.
- 2. running - תהליך בביצוע.
- 3. waiting - תהליך המתחיל להמתין לסיבוב שוב.
- 4. Termination - תהליך שהסתיים.

* cpu switch - מעבר ה-CPU בין תהליכים. מעבר ה-CPU מתבצע כאשר התהליך הנוכחי מסתיים או כאשר התהליך הנוכחי מתחיל להמתין לסיבוב שוב. מעבר ה-CPU מתבצע על ידי המערכת.

- job queue - תור של תהליכים מוכנים להיבצע על ידי ה-CPU.
- Ready queue - תור של תהליכים מוכנים להיבצע על ידי ה-CPU.
- Waiting queue - תור של תהליכים המתחילים להמתין לסיבוב שוב.

למטרות אחרות

* Context Switch - החלפת תוכנית - CPU של תוכנית אחרת
התוכנית החדשה צריכה להחזיר את המצב של התוכנית הקודמת, ולכן צריך לשמור על המצב של התוכנית הקודמת.

* Independent Process - תוכנית עצמאית - תוכנית שרצה בצורה עצמאית
כל תוכנית היא תוכנית עצמאית (אנחנו רוצים להבין).

* Cooperating Process - תוכנית שיתופית - תוכנית שרצה בצורה שיתופית
כל תוכנית היא תוכנית שיתופית.

* Shared Memory - זיכרון משותף - תוכנית שרצה בצורה שיתופית
כל תוכנית היא תוכנית שיתופית.

* Message Passing - מעבר מסר - תוכנית שרצה בצורה שיתופית
כל תוכנית היא תוכנית שיתופית.

* Kernel - ליבה - תוכנית שרצה בצורה שיתופית
כל תוכנית היא תוכנית שיתופית.

* Shared Library - ספרייה משותפת - תוכנית שרצה בצורה שיתופית
כל תוכנית היא תוכנית שיתופית.

* Shared Resource - משאב משותף - תוכנית שרצה בצורה שיתופית
כל תוכנית היא תוכנית שיתופית.

* Shared Data - נתונים משותפים - תוכנית שרצה בצורה שיתופית
כל תוכנית היא תוכנית שיתופית.

* Unbounded-buffer - בולדור מוגבל - תוכנית שרצה בצורה שיתופית
כל תוכנית היא תוכנית שיתופית.

* Bounded-buffer - בולדור מוגבל - תוכנית שרצה בצורה שיתופית
כל תוכנית היא תוכנית שיתופית.

* Thread - תהליך - תוכנית שרצה בצורה שיתופית
כל תוכנית היא תוכנית שיתופית.

* PC - נקודת קריאה, * תוכנית, * תוכנית
כל תוכנית היא תוכנית שיתופית.

* User - משתמש, * תוכנית, * תוכנית
כל תוכנית היא תוכנית שיתופית.

* Kernel - ליבה, * תוכנית, * תוכנית
כל תוכנית היא תוכנית שיתופית.

מקור המידע

על-פי-כך יש להבחין בין מודלים שונים של התנהגות מערכת. Many to One * - מודל שבו מספר תהליכים מתחברים אל תהליך אחד.

מקור המידע: על-פי-כך יש להבחין בין מודלים שונים של התנהגות מערכת. One to One * - מודל שבו תהליך אחד מתחבר אל תהליך אחד.

מקור המידע: על-פי-כך יש להבחין בין מודלים שונים של התנהגות מערכת. Many to Many * - מודל שבו מספר תהליכים מתחברים אל מספר תהליכים.

מקור המידע: על-פי-כך יש להבחין בין מודלים שונים של התנהגות מערכת. To-level Model * - מודל שבו יש מספר רמות (One to One, many to many).

* CPU Scheduling - מנגנון שבו ה-CPU מתחיל לנהל את התהליכים. Multi Programming (או תהליכים) - מודל שבו מספר תהליכים מתחברים אל ה-CPU.

מקור המידע: על-פי-כך יש להבחין בין מודלים שונים של התנהגות מערכת. Preemptive - מודל שבו תהליך אחד יכול להפסיק את תהליך אחר.

מקור המידע: על-פי-כך יש להבחין בין מודלים שונים של התנהגות מערכת. Non-preemptive - מודל שבו תהליך אחד לא יכול להפסיק את תהליך אחר.

מקור המידע: על-פי-כך יש להבחין בין מודלים שונים של התנהגות מערכת. Ready - מודל שבו תהליך אחד מוכן להתחיל.

מקור המידע: על-פי-כך יש להבחין בין מודלים שונים של התנהגות מערכת. Dispatcher * - מנגנון שבו ה-CPU מתחיל לנהל את התהליכים.

מקור המידע: על-פי-כך יש להבחין בין מודלים שונים של התנהגות מערכת. Context switch (או user mode) - מנגנון שבו מערכת מתחילה לנהל את התהליכים.

לוחות זמן

* First Come First served (FCFS) - לוחות זמן

לוחות זמן פשוטים. כל תהליך מתחיל כשמתקבל. אין תחלופה. Convoy Effect - תופעה שבה תהליכים ארוכים מונעים את תחילתם של תהליכים קצרים. Ready - תהליכים שמתחילים.

* Shortest job first Scheduling (SJF) - לוחות זמן

לוחות זמן שבהם מתחילים תהליכים קצרים קודם. Ready - תהליכים שמתחילים. FCFS - לוחות זמן פשוטים.

לוחות זמן שבהם מתחילים תהליכים קצרים קודם. Ready - תהליכים שמתחילים.

לוחות זמן שבהם מתחילים תהליכים קצרים קודם. Ready - תהליכים שמתחילים.

לוחות זמן שבהם מתחילים תהליכים קצרים קודם. Ready - תהליכים שמתחילים.

לוחות זמן שבהם מתחילים תהליכים קצרים קודם. Ready - תהליכים שמתחילים.

לוחות זמן שבהם מתחילים תהליכים קצרים קודם. Ready - תהליכים שמתחילים.

לוחות זמן שבהם מתחילים תהליכים קצרים קודם. Ready - תהליכים שמתחילים.

לוחות זמן שבהם מתחילים תהליכים קצרים קודם. Ready - תהליכים שמתחילים.

לוחות זמן שבהם מתחילים תהליכים קצרים קודם. Ready - תהליכים שמתחילים.

לוחות זמן שבהם מתחילים תהליכים קצרים קודם. Ready - תהליכים שמתחילים.

לוחות זמן שבהם מתחילים תהליכים קצרים קודם. Ready - תהליכים שמתחילים.

לוחות זמן שבהם מתחילים תהליכים קצרים קודם. Ready - תהליכים שמתחילים.

לוחות זמן שבהם מתחילים תהליכים קצרים קודם. Ready - תהליכים שמתחילים.

לוחות זמן שבהם מתחילים תהליכים קצרים קודם. Ready - תהליכים שמתחילים.

מדיניות התחנות

* Round Robin (RR) - מסתובב

התחנות: FCFS - מדיניות התחנות. קובע את סדרן של כל תחנה (Time Quantum) ומוזמן לתחנות. מדיניות התחנות מסתובבת, כל תחנה מקבלת זמן קבוע, וכל תחנה שמתחילה לרוץ מקבלת זמן קבוע. מדיניות התחנות מסתובבת.

התחנות מסתובבות. מדיניות התחנות מסתובבת. מדיניות התחנות מסתובבת.

- על התחנות מסתובבות.
- על התחנות מסתובבות.
- על התחנות מסתובבות.
- על התחנות מסתובבות.

Ready Queue - מדיניות התחנות. מדיניות התחנות מסתובבת.

* Multilevel Queue - מדיניות התחנות מסתובבת

מדיניות התחנות מסתובבת. מדיניות התחנות מסתובבת.

מדיניות התחנות מסתובבת. מדיניות התחנות מסתובבת.

* Multilevel Feedback Queue - מדיניות התחנות מסתובבת

מדיניות התחנות מסתובבת. מדיניות התחנות מסתובבת.

מבוא לתוכנית

* Dual Core Design - ארכיטקטורת שני ליבות

יש להבין את המודל של שני ליבות. המודל של שני ליבות הוא שיש לנו שתי ליבות, כל אחת עם זיכרון משותף משלה. המודל של שני ליבות הוא שיש לנו שתי ליבות, כל אחת עם זיכרון משותף משלה. המודל של שני ליבות הוא שיש לנו שתי ליבות, כל אחת עם זיכרון משותף משלה.

* Process Synchronization - סנכרון תהליכים

הסנכרון של תהליכים הוא תהליך של מניעת התנגשות בין תהליכים. הסנכרון של תהליכים הוא תהליך של מניעת התנגשות בין תהליכים. הסנכרון של תהליכים הוא תהליך של מניעת התנגשות בין תהליכים.

1. Counter - ספירה

הספירה היא תהליך של חישוב מספרים. הספירה היא תהליך של חישוב מספרים. הספירה היא תהליך של חישוב מספרים. הספירה היא תהליך של חישוב מספרים.

1. Mutual Exclusion - סגירת הדלת

הסגירת הדלת היא תהליך של מניעת כניסת תהליכים נוספים. הסגירת הדלת היא תהליך של מניעת כניסת תהליכים נוספים.

2. Progress - התקדמות

ההתקדמות היא תהליך של מניעת עיבודים. ההתקדמות היא תהליך של מניעת עיבודים. ההתקדמות היא תהליך של מניעת עיבודים.

3. Bounded Waiting - סגירת הדלת

הסגירת הדלת היא תהליך של מניעת כניסת תהליכים נוספים. הסגירת הדלת היא תהליך של מניעת כניסת תהליכים נוספים.

* Deadlock - עיבודים

העיבודים הם תהליכים של מניעת עיבודים. העיבודים הם תהליכים של מניעת עיבודים. העיבודים הם תהליכים של מניעת עיבודים.

מפרט המערכת

1. Preemptive - המערכת תפסק את המשימה הנוכחית ותתחיל את המשימה הבאה.

2. Non-preemptive - המערכת תמשיך להריץ את המשימה הנוכחית עד שתגיע לסוף.

(3) - Priority - המערכת תפסק את המשימה הנוכחית ותתחיל את המשימה בעלת הpriorities גבוה יותר.

4. Semaphore - מכשיר המאפשר למספר מסוים של תהליכים להיכנס לresource מסוים.

5. Lock - מכשיר המאפשר לתהליך אחד להיכנס לresource מסוים.

6. Monitor - מכשיר המאפשר לתהליך אחד להיכנס לresource מסוים.

7. Barrier - מכשיר המאפשר לתהליכים להמתין עד שכולם יגיעו לנקודה מסוימת.

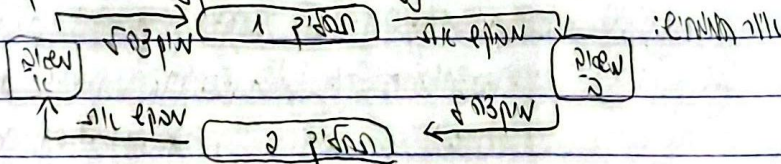
8. Read-write lock - מכשיר המאפשר לתהליכים לקרוא ולכתוב לresource מסוים.

9. Atomic - מכשיר המאפשר לתהליכים להיכנס לresource מסוים.

10. Cache coherence - מכשיר המאפשר לתהליכים להיכנס לresource מסוים.

11. Cache consistency - מכשיר המאפשר לתהליכים להיכנס לresource מסוים.

12. Cache coherency - מכשיר המאפשר לתהליכים להיכנס לresource מסוים.



* Starvation - תהליך המתחיל להמתין לresource מסוים ולא מקבל אותו.

* Priority inversion - יתרון לpriority גבוה יותר על פני priority נמוך יותר.

13. Priority inversion - יתרון לpriority גבוה יותר על פני priority נמוך יותר.

מבטות רשת

* Readers Writers Problem - קבוצת כותבים וקבוצת קוראים על אותו נתון

הקוראים יכולים לקרוא בו-זמנית

אך הכותבים חייבים לכתוב בודדים

אם יש כותב אחד או יותר, אין קוראים

אם יש קוראים אחד או יותר, אין כותבים

אם יש כותב אחד או יותר, אין קוראים

אם יש קוראים אחד או יותר, אין כותבים

אם יש כותב אחד או יותר, אין קוראים

אם יש קוראים אחד או יותר, אין כותבים

- 3 סוגי mutex: read, write, rw

* rw-mutex - mutex עם 2 מצבים: read ו-write

* mutex - mutex עם מצב אחד: lock

* read-count - מניין מספר הקוראים

אם יש קוראים אחד או יותר, אין כותבים

אם יש כותב אחד או יותר, אין קוראים

signal (rw-mutex)

אם יש קוראים אחד או יותר, אין כותבים

אם יש כותב אחד או יותר, אין קוראים

אם יש קוראים אחד או יותר, אין כותבים

- 3 סוגי mutex: read, write, rw

* semaphore - semaphore עם 2 מצבים: read ו-write

אם יש קוראים אחד או יותר, אין כותבים

אם יש כותב אחד או יותר, אין קוראים

אם יש קוראים אחד או יותר, אין כותבים

אם יש כותב אחד או יותר, אין קוראים

- 3 סוגי mutex: read, write, rw

אם יש קוראים אחד או יותר, אין כותבים

אם יש כותב אחד או יותר, אין קוראים

אם יש קוראים אחד או יותר, אין כותבים

- 3 סוגי mutex: read, write, rw

אם יש קוראים אחד או יותר, אין כותבים

מכרזות

Deadlocks * (החלטה) - קבוצה של תהליכים המחזיקים משאבים זהים וכל אחד מהם מחכה למשאבים שנתחבט על ידי תהליכים אחרים.

תהליכים אלו יכולים להימנע על ידי שימוש במנגנוני מניעה או הימנעות. לדוגמה, שימוש ב-Mutex.

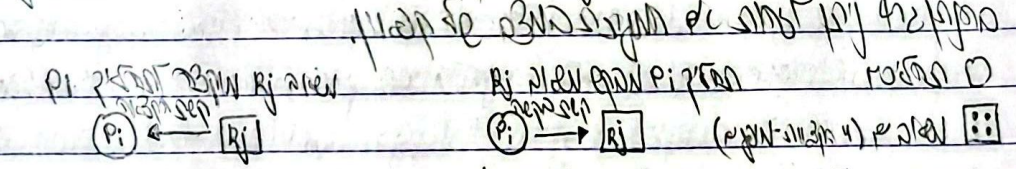
Deadlock Characterization

- 1. Mutual Exclusion - משאבים אינם ניתנים לשימוש משותף.
- 2. Hold and wait - תהליך מחזיק משאבים וצריך להחזיר אותם כדי שיתחבט על ידי תהליכים אחרים.
- 3. No preemption - משאבים אינם ניתנים להחזרה מאלו שחזקו אותם.

Circular wait

תהליכים P_1, P_2, \dots, P_n מחזיקים משאבים זהים. P_1 מחכה למשאבים של P_2 , P_2 מחכה למשאבים של P_3 , ..., P_n מחכה למשאבים של P_1 .

System Resource Allocation Graph *



החלטה על מתי תהליכים יקבלו משאבים, על פי סדר הקבלה. זה יכול להוביל למצב של מחצית.

! חלק מהמשאבים יישאר חסומים (dead) ולא יוכלו להיחזר.

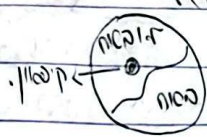
Deadlock avoidance

- 1. מניעה/הימנעות - שימוש במנגנונים כמו Banker's algorithm.
- 2. מניעה - שימוש במנגנונים כמו semaphore.
- 3. הימנעות - שימוש במנגנונים כמו priority inheritance.

מנגנוני אבטלה

* Safe State - מצב בו כל התהליכים יוכלו להיגמר בלי שום בעיה. כלומר, כל התהליכים יוכלו להשיג את משאביהם מבלי שיש צורך להמתין למשאבים אחרים. זהו מצב בו אין צורך להמתין למשאבים אחרים.

המנגנון הבסיסי ביותר לזיהוי מצב בטוח הוא אלגוריתם הבנקאי (The Banker's Algorithm). אלגוריתם זה מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה.



על המנגנון "מס" להודיע על מצב בטוח (המנגנון הבסיסי ביותר).
 על המנגנון להודיע על מצב בטוח ויכולת לזכות במשאבים.
 המנגנון הבסיסי ביותר לזיהוי מצב בטוח הוא אלגוריתם הבנקאי.

* The Banker's Algorithm - המנגנון הבסיסי ביותר לזיהוי מצב בטוח. המנגנון מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה.

המנגנון מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה. המנגנון מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה.

המנגנון מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה. המנגנון מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה.

המנגנון מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה. המנגנון מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה.

המנגנון מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה. המנגנון מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה.

* Detection Algorithm - המנגנון הבסיסי ביותר לזיהוי מצב של דיאדלוק. המנגנון מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה.

המנגנון מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה. המנגנון מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה.

המנגנון מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה. המנגנון מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה.

המנגנון מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה. המנגנון מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה.

a. זיהוי מצב של דיאדלוק - instances - המנגנון מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה.

המנגנון מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה. המנגנון מבטיח שכל התהליכים יוכלו להיגמר בלי שום בעיה.

מחזורי זיכרון

הקצאת זיכרון (Memory Allocation) היא תהליך של הקצאת זיכרון לפרוייקט. זהו תהליך שבו המערכת מודעת לזיכרון שיש להקצות לפרוייקט ולקצות אותו בצורה יעילה.

ישנן שתי שיטות עיקריות להקצאת זיכרון: Static Allocation ו- Dynamic Allocation.
- Static Allocation: הקצאת זיכרון מתבצעת מראש על ידי המפתח, ללא תלות בגודל הפרוייקט.
- Dynamic Allocation: הקצאת זיכרון מתבצעת בזמן ריצת הפרוייקט, על ידי המערכת או המשתמש.

* Cache - זיכרון מהיר הממוקם בין המעבד לזיכרון הראשי. מטרתו היא לשפר את מהירות הגישה לנתונים הנמצאים בו. Cache הוא זיכרון קטן מאוד, אך מהיר מאוד.

* Logical Address - כתובת זיכרון לוגית. היא כתובת זיכרון שנוצרת על ידי המערכת, ולא כתובת זיכרון פיזית. Logical Address היא כתובת זיכרון שנוצרת על ידי המערכת, ולא כתובת זיכרון פיזית.

base - כתובת זיכרון בסיסית. היא כתובת זיכרון שנוצרת על ידי המערכת, ולא כתובת זיכרון פיזית.
limit - גודל זיכרון. הוא גודל זיכרון שנוצרת על ידי המערכת, ולא גודל זיכרון פיזית.
base + limit = גודל זיכרון. הוא גודל זיכרון שנוצרת על ידי המערכת, ולא גודל זיכרון פיזית.

* Memory Management Unit (MMU) - יחידת ניהול זיכרון. היא יחידת זיכרון שנוצרת על ידי המערכת, ולא יחידת זיכרון פיזית. מטרתה היא לנהל את זיכרון המערכת בצורה יעילה.

* Backing store - זיכרון אחסון. הוא זיכרון שנוצרת על ידי המערכת, ולא זיכרון פיזית. מטרתו היא לשמור את הנתונים שנוצרו על ידי המערכת, ולא לשמור את הנתונים שנוצרו על ידי המערכת.

* Contiguous Allocation - הקצאת זיכרון רציפה. היא הקצאת זיכרון שנוצרת על ידי המערכת, ולא הקצאת זיכרון פיזית. מטרתה היא להקצות זיכרון רציף לפרוייקט.

* Multiple partition memory - זיכרון מרוב חלקים. הוא זיכרון שנוצרת על ידי המערכת, ולא זיכרון פיזית. מטרתו היא להקצות זיכרון מרוב חלקים לפרוייקט.

* First fit - שיטת הקצאת זיכרון ראשונה. היא שיטת הקצאת זיכרון שנוצרת על ידי המערכת, ולא שיטת הקצאת זיכרון פיזית. מטרתה היא להקצות זיכרון ראשונה לפרוייקט.

* Best fit - שיטת הקצאת זיכרון הטובה ביותר. היא שיטת הקצאת זיכרון שנוצרת על ידי המערכת, ולא שיטת הקצאת זיכרון פיזית. מטרתה היא להקצות זיכרון הטובה ביותר לפרוייקט.

* Worst fit - שיטת הקצאת זיכרון הגרועה ביותר. היא שיטת הקצאת זיכרון שנוצרת על ידי המערכת, ולא שיטת הקצאת זיכרון פיזית. מטרתה היא להקצות זיכרון הגרועה ביותר לפרוייקט.

מנגנון זיכרון

invalid ee zero - זיכרון שגוי (Error) - זיכרון לא תקין

page fault - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

$$EAT = (1-p) * Memory Access + p * (swapout + swap in) \quad : \text{זיכרון שגוי}$$

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

זיכרון שגוי - זיכרון לא תקין

מקורות מידע

- Second Chance (clock) Page Replacement Algorithm *

FFFO עניין - מיון לפי זמן: קבוע א זמן עתידית
 • זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש
 • זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש
 • זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש
 • זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש
 • זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש
 • זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש
 • זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש

- Counting Algorithm *

page 55 פירוש - זמן פירוש - זמן פירוש - זמן פירוש
 • זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש

• Least Frequently Used (LFU) Algorithm - זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש

• Most Frequently Used (MFU) Algorithm - זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש

- Fixed Allocation *

• Equal allocation - זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש

• Proportional allocation - זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש

• Priority Allocation - זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש

• Global replacement - זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש

• Local replacement - זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש

• Trashing - זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש

• זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש

• page fault - זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש

• זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש

• זמן פירוש - זמן פירוש - זמן פירוש - זמן פירוש

מבנה מערכת

- * Attributes - תכונות של קובץ (PCB, size, type, name, etc.)
- * Partitions - חלוקה של דיסקים לפרטים (primary, secondary, etc.)
- * Path - כתובת של קובץ
- * Absolute Path - כתובת מוחלטת
- * Relative Path - כתובת יחסית
- * dangling pointer - מצביע תלול
- * Mounting - התקנת מערכת קובצים
- * logical file system - מערכת קובצים לוגית
- * File organization module - מודול ארגון קובצים
- * Basic file system - מערכת קובצים בסיסית
- * Device driver - דרייבר של מכשיר
- * File Control Block (FCB) - בלוק בקרה של קובץ
- * Contiguous - קובץ רציף
- * Indexed Allocation - 할당 ממופת
- * Free Space Management - ניהול חלל חופשי
- * Recovery - שחזור
- * Consistency Checker - בודק עקביות
- * Back up - גיבוי

מבנה מערכת

המערכת מורכבת מרכיבים שונים, המיושמים על ידי מערכת ההפעלה. המערכת מורכבת מרכיבים שונים, המיושמים על ידי מערכת ההפעלה. המערכת מורכבת מרכיבים שונים, המיושמים על ידי מערכת ההפעלה.

*** Device Driver** - מודול המאפשר למערכת להשתמש בציוד חיצוני. המודול מיושם על ידי מערכת ההפעלה.

המערכת מורכבת מרכיבים שונים, המיושמים על ידי מערכת ההפעלה. המערכת מורכבת מרכיבים שונים, המיושמים על ידי מערכת ההפעלה.

*** Index Allocation** - מודול המאפשר למערכת להשתמש בציוד חיצוני. המודול מיושם על ידי מערכת ההפעלה.

המערכת מורכבת מרכיבים שונים, המיושמים על ידי מערכת ההפעלה. המערכת מורכבת מרכיבים שונים, המיושמים על ידי מערכת ההפעלה.

המערכת מורכבת מרכיבים שונים, המיושמים על ידי מערכת ההפעלה. המערכת מורכבת מרכיבים שונים, המיושמים על ידי מערכת ההפעלה.

המערכת מורכבת מרכיבים שונים, המיושמים על ידי מערכת ההפעלה. המערכת מורכבת מרכיבים שונים, המיושמים על ידי מערכת ההפעלה.

*** RAID** - מערכת אחסון נתונים המורכבת מרשתת דיסקים. המערכת מיושמת על ידי מערכת ההפעלה.

המערכת מורכבת מרכיבים שונים, המיושמים על ידי מערכת ההפעלה. המערכת מורכבת מרכיבים שונים, המיושמים על ידי מערכת ההפעלה.

RAID 0 - מערכת אחסון נתונים המורכבת מרשתת דיסקים. המערכת מיושמת על ידי מערכת ההפעלה.

המערכת מורכבת מרכיבים שונים, המיושמים על ידי מערכת ההפעלה. המערכת מורכבת מרכיבים שונים, המיושמים על ידי מערכת ההפעלה.

RAID 1 - מערכת אחסון נתונים המורכבת מרשתת דיסקים. המערכת מיושמת על ידי מערכת ההפעלה.

המערכת מורכבת מרכיבים שונים, המיושמים על ידי מערכת ההפעלה. המערכת מורכבת מרכיבים שונים, המיושמים על ידי מערכת ההפעלה.

RAID 5 - מערכת אחסון נתונים המורכבת מרשתת דיסקים. המערכת מיושמת על ידי מערכת ההפעלה.

המערכת מורכבת מרכיבים שונים, המיושמים על ידי מערכת ההפעלה. המערכת מורכבת מרכיבים שונים, המיושמים על ידי מערכת ההפעלה.

*** גודל נתונים:** 1 byte = 8 bits, KB = 2¹⁰ = 1024B, MB = 2²⁰ = 1024KB, GB = 2³⁰ = 1024MB, TB = 2⁴⁰ = 1024GB

מבוא לתוכנית - מבוא

- * cmd - ממשק שורת פקודות
- * Registry - רשימת הגדרות המערכת
- * Process Explorer - תוכנית המציגה את כל תהליכי המערכת
- * Multiprocessors - מערכת עם מספר תהליכים

- * Symmetric - סימטרית
- * Asymmetric - אסימטרית
- * User Mode - מצב משתמש
- * kernel mode - מצב גרעין

- * interrupt - הפרעה
- * virtualization - וירטואליזציה

- * shell - קליפה
- * ls - רשימת קבצים
- * ls -l filename - רשימת פרטי קובץ
- * pwd - מיקום התיקיה הנוכחית
- * cd - מעבר לתיקיה
- * mkdir directory - יצירת תיקיה
- * touch filename - יצירת קובץ
- * pic filename - יצירת קובץ פיקסל
- * top - מציג את תהליכי המערכת
- * ps -ef - מציג את תהליכי המערכת
- * ps 255 - מציג את תהליכי המערכת
- * kill 7606 - מציג את תהליכי המערכת
- * bg - מציג את תהליכי המערכת
- * jobs - מציג את תהליכי המערכת

מבוא - תורת המחשבים

gcc -o first first.c

* gcc תכנתה ויש לה שם של gcc

היא לא תבצע את הפיקצורה הראשונה (הראשונה) של הקובץ הראשון.

היא תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

היא תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

* wait תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

(היא לא תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.)

היא תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

! כמות המילים של הקובץ הראשון (הראשונה) של הקובץ הראשון.

היא תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

היא תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

היא תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

היא תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

* exec תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

היא תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

* PCB - תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

היא תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

היא תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

* wait - תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

היא תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

היא תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

היא תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

היא תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

! היא תבצע את הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

- סולם של מחשבים: So - sleep - זמן של הפיקצורה הראשונה של הקובץ הראשון.

0 - מחשבים של מחשבים, T (wait) - זמן של הפיקצורה הראשונה של הקובץ הראשון.

0 - מחשבים של מחשבים, R (RAM) - זמן של הפיקצורה הראשונה של הקובץ הראשון.

- זמן של הפיקצורה הראשונה של הקובץ הראשון (הראשונה) של הקובץ הראשון.

מבנה מערכת - Threads

* Threads - processes ו-threads הם שני מושגים שונים. processes הם תהליכים המיושמים על ידי המערכת, ו-threads הם חלקים קטנים יותר של תהליכים. כל תהליך יכול להכיל מספר threads.

* Process - תהליך הוא יחידת העבודה הבסיסית של מערכת ההפעלה. הוא כולל מטא-נתונים (metadata) כגון זיכרון, קבצי נתונים וכו'.

* Thread - קוץ הוא יחידת העבודה הבסיסית של תהליך. הוא יכול להשתמש במשאבים של התהליך שלו, אך לא במשאבים של תהליכים אחרים. כל תהליך יכול להכיל מספר threads. מטרת ה-threads היא להקטין את זמן הריצה של התהליך.

! Kernel - מערכת ההפעלה היא אחראית על יצירת ה-threads. היא מאפשרת לתהליכים ליצור threads משלהם. ה-threads הם חלק מהמערכת, והמערכת היא אחראית על ניהולם.

! Thread - קוץ הוא יחידת העבודה הבסיסית של תהליך. הוא יכול להשתמש במשאבים של התהליך שלו, אך לא במשאבים של תהליכים אחרים. כל תהליך יכול להכיל מספר threads.

* Thread - קוץ הוא יחידת העבודה הבסיסית של תהליך. הוא יכול להשתמש במשאבים של התהליך שלו, אך לא במשאבים של תהליכים אחרים. כל תהליך יכול להכיל מספר threads.

* Thread - קוץ הוא יחידת העבודה הבסיסית של תהליך. הוא יכול להשתמש במשאבים של התהליך שלו, אך לא במשאבים של תהליכים אחרים. כל תהליך יכול להכיל מספר threads.

* Thread - קוץ הוא יחידת העבודה הבסיסית של תהליך. הוא יכול להשתמש במשאבים של התהליך שלו, אך לא במשאבים של תהליכים אחרים. כל תהליך יכול להכיל מספר threads.

* Thread - קוץ הוא יחידת העבודה הבסיסית של תהליך. הוא יכול להשתמש במשאבים של התהליך שלו, אך לא במשאבים של תהליכים אחרים. כל תהליך יכול להכיל מספר threads.

* Thread - קוץ הוא יחידת העבודה הבסיסית של תהליך. הוא יכול להשתמש במשאבים של התהליך שלו, אך לא במשאבים של תהליכים אחרים. כל תהליך יכול להכיל מספר threads.

מבוא לתורת הקבוצות

* SKW Lock - כל נכנס לחדר קודם שיש לו מפתח. כל מי שיש לו מפתח יכול להיכנס לחדר.

- כל מי שיש לו מפתח יכול להיכנס לחדר, אבל מי שיש לו מפתח יכול להיכנס לחדר.

* בעיה של מפתח - יש לחדר מפתח, אבל מי שיש לו מפתח יכול להיכנס לחדר.

- כל מי שיש לו מפתח יכול להיכנס לחדר, אבל מי שיש לו מפתח יכול להיכנס לחדר.

- מפתח - כל מי שיש לו מפתח יכול להיכנס לחדר, אבל מי שיש לו מפתח יכול להיכנס לחדר.

* אלגוריתם ה-Bakery (Bakery Algorithm) - כל מי שיש לו מפתח יכול להיכנס לחדר.

- כל מי שיש לו מפתח יכול להיכנס לחדר, אבל מי שיש לו מפתח יכול להיכנס לחדר.

- מפתח - כל מי שיש לו מפתח יכול להיכנס לחדר, אבל מי שיש לו מפתח יכול להיכנס לחדר.

- כל מי שיש לו מפתח יכול להיכנס לחדר, אבל מי שיש לו מפתח יכול להיכנס לחדר.

1. מפתח - כל מי שיש לו מפתח יכול להיכנס לחדר, אבל מי שיש לו מפתח יכול להיכנס לחדר.

2. מפתח - כל מי שיש לו מפתח יכול להיכנס לחדר, אבל מי שיש לו מפתח יכול להיכנס לחדר.

3. מפתח - כל מי שיש לו מפתח יכול להיכנס לחדר, אבל מי שיש לו מפתח יכול להיכנס לחדר.

4. מפתח - כל מי שיש לו מפתח יכול להיכנס לחדר, אבל מי שיש לו מפתח יכול להיכנס לחדר.

* בעיה של מפתח - כל מי שיש לו מפתח יכול להיכנס לחדר, אבל מי שיש לו מפתח יכול להיכנס לחדר.

1. מפתח - כל מי שיש לו מפתח יכול להיכנס לחדר, אבל מי שיש לו מפתח יכול להיכנס לחדר.

2. מפתח - כל מי שיש לו מפתח יכול להיכנס לחדר, אבל מי שיש לו מפתח יכול להיכנס לחדר.

3. מפתח - כל מי שיש לו מפתח יכול להיכנס לחדר, אבל מי שיש לו מפתח יכול להיכנס לחדר.

4. מפתח - כל מי שיש לו מפתח יכול להיכנס לחדר, אבל מי שיש לו מפתח יכול להיכנס לחדר.

מבוא לתורת המשפט

* כל משפט מתמטי הוא או נכון או שגוי.
 * כל משפט מתמטי הוא או נכון או שגוי.
 * כל משפט מתמטי הוא או נכון או שגוי.

* כל משפט מתמטי הוא או נכון או שגוי?

המשפט הוא נכון או שגוי.
 המשפט הוא נכון או שגוי.

המשפט הוא נכון או שגוי.
 המשפט הוא נכון או שגוי.

* כל משפט מתמטי הוא או נכון או שגוי.

המשפט הוא נכון או שגוי.

המשפט הוא נכון או שגוי.

המשפט הוא נכון או שגוי.

המשפט הוא נכון או שגוי.

* כל משפט מתמטי הוא או נכון או שגוי.

המשפט הוא נכון או שגוי.

* כל משפט מתמטי הוא או נכון או שגוי.

המשפט הוא נכון או שגוי.

המשפט הוא נכון או שגוי.

המשפט הוא נכון או שגוי.

המשפט הוא נכון או שגוי.

המשפט הוא נכון או שגוי.

המשפט הוא נכון או שגוי.

המשפט הוא נכון או שגוי.

המשפט הוא נכון או שגוי.

המשפט הוא נכון או שגוי.

המשפט הוא נכון או שגוי.

* כל משפט מתמטי הוא או נכון או שגוי.

המשפט הוא נכון או שגוי.

המשפט הוא נכון או שגוי.

מנעול - semaphore

הוא מנעול סימבולי, כלומר הוא מנעול שיש לו מצב של "נעול" או "פתוח".
הוא מיושם באמצעות מבנה הנתונים הבא:

* semaphore - מבנה הנתונים הבא:

waiters (מספר המתחייבים) ו- mutex (מנעול)

המתחייבים יתחייבו את המנעול באמצעות waiters.

המנעול יתחייב את המתחייבים באמצעות mutex.

* Spooler - מבנה הנתונים הבא:

הוא מבנה הנתונים הבא:

• mutex (מנעול) ו- spooler (מבנה הנתונים)

המבנה יתחייב את המנעול באמצעות mutex.

המנעול יתחייב את המבנה באמצעות spooler.

* Mutex - מבנה הנתונים הבא:

הוא מבנה הנתונים הבא:

הוא מבנה הנתונים הבא:

* semaphore - מבנה הנתונים הבא:

הוא מבנה הנתונים הבא:

הוא מבנה הנתונים הבא:

הוא מבנה הנתונים הבא:

הוא מבנה הנתונים הבא:

הוא מבנה הנתונים הבא:

הוא מבנה הנתונים הבא:

הוא מבנה הנתונים הבא:

הוא מבנה הנתונים הבא:

הוא מבנה הנתונים הבא:

הוא מבנה הנתונים הבא:

הוא מבנה הנתונים הבא:

הוא מבנה הנתונים הבא:

הוא מבנה הנתונים הבא:

* mutex - מבנה הנתונים הבא:

הוא מבנה הנתונים הבא:

הוא מבנה הנתונים הבא:

מבטאים את המידע

* קובץ נתונים - מיון נתונים לפי סדר מסוים של נתונים על מנת להקטין את זמן הריצה.
 * קובץ נתונים - מיון נתונים לפי סדר מסוים של נתונים על מנת להקטין את זמן הריצה.
 * קובץ נתונים - מיון נתונים לפי סדר מסוים של נתונים על מנת להקטין את זמן הריצה.
 * קובץ נתונים - מיון נתונים לפי סדר מסוים של נתונים על מנת להקטין את זמן הריצה.

* Node Table - מיון נתונים לפי סדר מסוים של נתונים על מנת להקטין את זמן הריצה.
 * Node Table - מיון נתונים לפי סדר מסוים של נתונים על מנת להקטין את זמן הריצה.
 * Node Table - מיון נתונים לפי סדר מסוים של נתונים על מנת להקטין את זמן הריצה.

* Node Table - מיון נתונים לפי סדר מסוים של נתונים על מנת להקטין את זמן הריצה.
 * Node Table - מיון נתונים לפי סדר מסוים של נתונים על מנת להקטין את זמן הריצה.

* Node Table - מיון נתונים לפי סדר מסוים של נתונים על מנת להקטין את זמן הריצה.
 * Node Table - מיון נתונים לפי סדר מסוים של נתונים על מנת להקטין את זמן הריצה.

* Node Table - מיון נתונים לפי סדר מסוים של נתונים על מנת להקטין את זמן הריצה.
 * Node Table - מיון נתונים לפי סדר מסוים של נתונים על מנת להקטין את זמן הריצה.

* Node Table - מיון נתונים לפי סדר מסוים של נתונים על מנת להקטין את זמן הריצה.
 * Node Table - מיון נתונים לפי סדר מסוים של נתונים על מנת להקטין את זמן הריצה.

עמוד המפתח - מפתח

* המפתח של המפתח הוא למעשה המפתח עצמו
 * כל מפתח הוא מפתח של מפתח אחר
 * המפתח של המפתח הוא למעשה המפתח עצמו

* המפתח של המפתח הוא למעשה המפתח עצמו
 * המפתח של המפתח הוא למעשה המפתח עצמו
 * המפתח של המפתח הוא למעשה המפתח עצמו

42 ביט
 $4MB = 4 \cdot 2^{20} B = 2^{22} B \Rightarrow$ כמות 22 סיביות
 $\Rightarrow 32 - 22 = 10$ סיביות ל-10 סיביות

* המפתח של המפתח הוא למעשה המפתח עצמו
 * המפתח של המפתח הוא למעשה המפתח עצמו

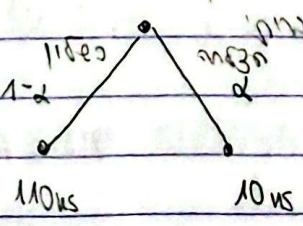
| | | | | | | | |
|------|------|------|------|------|------|------|------|
| A | B | C | D | E | F | 4 | 8 |
| 1010 | 1011 | 1100 | 1101 | 1110 | 1111 | 0100 | 1000 |

סיביות 10 ל-10 סיביות
 סיביות 22 ל-22 סיביות
 (page, offset) מ-10 סיביות
 (frame, offset) מ-22 סיביות

* המפתח של המפתח הוא למעשה המפתח עצמו
 * המפתח של המפתח הוא למעשה המפתח עצמו
 $4MB * 8 + DEF48 = 32 \cdot 2^{20} + DEF48 = 2^5 \cdot 2^{20} = 2^{25} + DEF48$
 $2^{25} + 913224 = 3MB = 7,656 = 20DEF48$

* המפתח של המפתח הוא למעשה המפתח עצמו
 * המפתח של המפתח הוא למעשה המפתח עצמו
 * המפתח של המפתח הוא למעשה המפתח עצמו

EAT = (1-α) * 10ns + α * 11ns
 $\Rightarrow (1-\alpha) \cdot 10 + \alpha \cdot 11 = 11$
 $\Rightarrow 10 - 10\alpha + 11\alpha = 11$
 $\Rightarrow 10 = 100\alpha \Rightarrow \alpha = 0.99$



שיעורי תוכנית - פירוק משימה

מבלי לפרט פקד, 7,200rpm וכו' נראה שישנה קוץ קצת בין: פירוק משימה *
 פירוק משימה, הפירוק/מסד ע"י שריר פירוק נעשה באמצעות (פירוק משימה) פירוק משימה
 פירוק משימה פירוק משימה (פירוק משימה) פירוק משימה

$\frac{1}{7200} \cdot 60 \cdot 10^6$

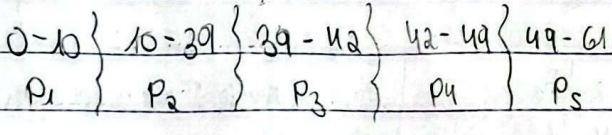
פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה

פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה

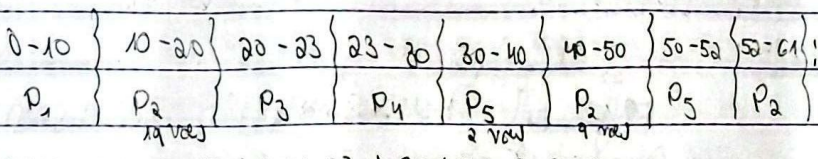
| Job | Burst Time |
|-----|------------|
| 1 | 10 |
| 2 | 29 |
| 3 | 3 |
| 4 | 7 |
| 5 | 12 |

200 מ"ס בין מירון מירון מירון מירון מירון
 ? פירוק משימה (פירוק משימה) פירוק משימה

FCFS (1)
 RR with q = 10 (2)
 SJF (2)
 RR with q = 5 (3)

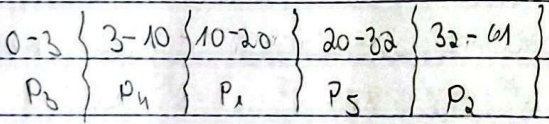


FCFS פירוק משימה
 $200 \text{ מ"ס} = \frac{10 + 39 + 42 + 49 + 61}{5} = 40.2$



q=10 פירוק משימה פירוק משימה

פירוק משימה פירוק משימה = $\frac{10 + 61 + 23 + 30 + 52}{5} = 35.2$



200 מ"ס = $\frac{3 + 10 + 20 + 32 + 61}{5} = 25.2$ SJF פירוק משימה

פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה
 פירוק משימה (SJF, SJF) פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה
 פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה

| Process | Burst time | Arrival | SJF |
|----------------|------------|---------|--|
| P ₁ | 7.5 | 0 | 0 } 1 } 2 } 3 } 4 } 5 } 6 } 7 } 8 } 9 } 10 } 11 } |
| P ₂ | 1 | 2 | P ₁ } P ₁ } P ₂ } P ₃ } P ₃ } P ₃ } P ₃ } P ₁ } P ₁ } P ₁ } P ₁ } |
| P ₃ | 4 | 3 | 1 2 3 4 |

context switch פירוק משימה פירוק משימה

Max-Allocation = Need פירוק משימה פירוק משימה פירוק משימה פירוק משימה פירוק משימה

שיעור 10 - פונקציות

- * זמן המתנה של המערכת - waiting time
- זמן סיבוב - turnaround time
- זמן תגובת המערכת - response time
- * Mutual Exclusion - זמן סגור
- * progress - זמן פתוח
- * Bounded waiting - זמן סגור מוגבל

- seek time - זמן חיפוש
- rotation time - זמן סיבוב
- transfer time - זמן העברה
- * זמן ממוצע: $\frac{1}{n} \sum_{i=1}^n (t_i + r_i)$

מספר המערכות: 10, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100
 זמן סיבוב: 100ms
 זמן חיפוש: 5ms
 זמן העברה: 1ms

→ SCAN (זמן סיבוב מוגבל)

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 10 | 20 | 25 | 45 | 90 | 80 | 70 | 60 | 50 | 40 |
|----|----|----|----|----|----|----|----|----|----|

הזמן הכולל של המערכת הוא סכום הזמנים של כל המערכות. במקרה של SCAN, הזמן הכולל הוא סכום הזמנים של כל המערכות. במקרה של C-SCAN, הזמן הכולל הוא סכום הזמנים של כל המערכות.

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 10 | 20 | 25 | 45 | 80 | 70 | 60 | 50 |
|----|----|----|----|----|----|----|----|

הזמן הכולל של המערכת הוא 138ms. הזמן הכולל של המערכת הוא 138ms.

$$138 - 130 = 8$$

מבנה זיכרון - מיון

* מבנה זיכרון: זיכרון ממוין - זיכרון ממוין

4KB בלי - 4KB בלי (מיון ממוין) - זיכרון ממוין

זיכרון ממוין - זיכרון ממוין

זיכרון ממוין - זיכרון ממוין

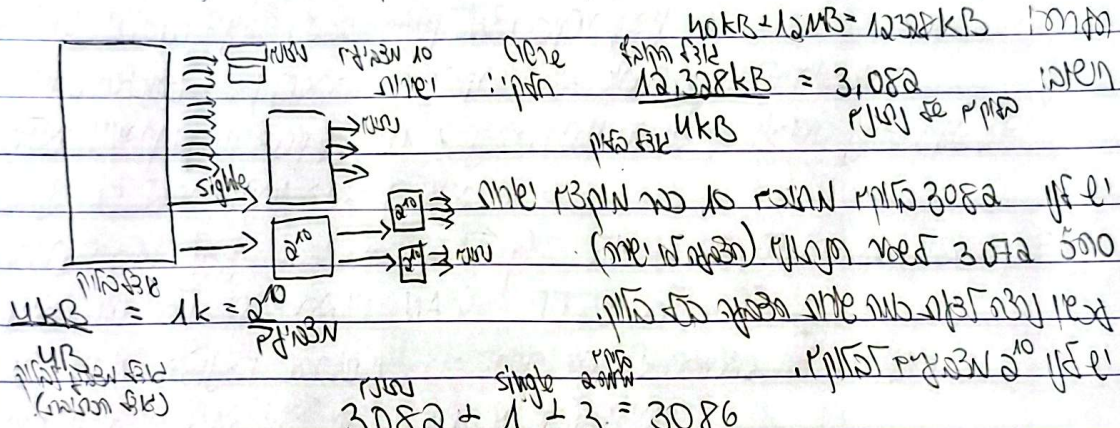
זיכרון ממוין - זיכרון ממוין

זיכרון ממוין - זיכרון ממוין

זיכרון ממוין - זיכרון ממוין

זיכרון ממוין - זיכרון ממוין

זיכרון ממוין - זיכרון ממוין



$10 + 2^{10} + 2^{10} \times 2^{10} + \dots$
 זיכרון ממוין - זיכרון ממוין

* מבנה זיכרון: זיכרון ממוין

זיכרון ממוין (frames) - זיכרון ממוין

זיכרון ממוין - זיכרון ממוין

זיכרון ממוין - זיכרון ממוין

$4MB = 4 \times 2^{20} = 4096$
 $4MB = 2048$
 $4096 + 2048 + 1024 + 512 = 7680$

זיכרון ממוין - זיכרון ממוין

זיכרון ממוין - זיכרון ממוין

זיכרון ממוין - זיכרון ממוין

זיכרון ממוין - זיכרון ממוין