

## שאלה 1

Operation	Protected/Non	Explanation
Set value of computer clock.	Protected	שינוי של שעון המערכת יכול לייצר בעיות אבטחה, ואי עקביות בתוכניות שתלויות בשעון (כלומר משפיע על הרבה יותר מסתם שינוי שעון)
Read the computer clock.	Non-Protected	יש תוכניות אשר תלויות בשעון המערכת לכן רלוונטי שיוכלו לקרוא אותו
Make intensive calculations.	Non-Protected	חלק עיקרי של תוכניות ועקרונית לא מייצרים בעיה למערכת
Read the memory of other processes.	Protected	יכול לייצר בעיות אבטחה, לדוגמה תוכניות יכולות לדלות סממאות/מידע אישי שלא בהכרח אמורות לדעת
Issue a trap/exception instruction.	Protected	יכול לגרום לתוכניות לייצר בעיות במערכת והניהול התקין שלה
Block all interrupts in the system.	Protected	יכול לפגוע ביציבות המערכת, להפריע למערכות זמן-אמת
Switch from user to kernel mode (change the mode-bit).	Protected	יכול לאפשר לתוכניות לעקוף את מנגנוני האבטחה של המערכת ולאפשר גישה בלתי נשלטת למקומות רגישים במערכת
Switch from kernel to user mode (change the mode-bit).	Non-Protected	כל שינוי במצב המשתמש צריך להתנהל על ידי מערכת ההפעלה ולא על ידי תוכניות אחרות, אך למעבר כזה אין פונציאל לפגוע במערכת.
Read the keyboard input.	Protected	קריאה ישירות מהמקלדת יכולה לגרום Keyloggers לבעיות אבטחה כמו
Read the mouse input.	Protected	בדומה למקלדת, מאפשר לעקוב אחרי המשתמש והפעולות שלו
Making a 'beep' sound .	Non-Protected	בדרך כלל אין השפעה על המערכת ומשמש לקבלת פידבק בהקשרים מסויימים
Read current user name (not the password)	Non-Protected	אין בעיה, מאפשר למערכת לייצר סביבה מותאמת אישית למשתמש
Access the hard disk drive (HDD) for writing.	Protected	יכול לגרום להתקנת תוכנות זדוניות או אפילו מחיקה של מערכת ההפעלה
Access the Wi-Fi hardware for sending packets.	Non-Protected	RCE יכול לגרום לבעיות אבטחה למשל attack
Controlling the keyboard status LEDs: CAPS-LOCK & NUM-LOCK	Non-Protected	מאפשר לתת פידבק על מצב המקלדת, לא מתאפשרת פגיעה במערכת
Controlling the microphone (recording)	Protected	בעיות אבטחה, מעקב אחרי המשתמש
Shutting down the computer	Protected	יכול לגרום לבעיות אבטחה או מחיקה של מידע
Putting a pixel on screen at (x,y)	Non-Protected	אומנם יכול לאפשר יצירה של אובייקטים בעייתיים על המסך, אך באופן כללי משמש להצגה גרפית של מערכות ולכן בכלליות אין בעיה
Read the number of processes in the system (how many processes are currently running, with no further information)	Non-Protected	מאפשר לספק מידע על המערכת ללא פגיעה בה
(Smartphones OS): reading the battery charge level.	Non-Protected	מאפשר לספק מידע על המשתמש ללא פגיעה בביטחון מידע

## שאלה 2

א. טבלת האינטרפטים היא סוג של מבנה נתונים שמערכת ההפעלה משתמשת בו בשביל לנהל אינטרפטים. כל כניסה בטבלה מתאימה לאינטרפט ספציפי ומצביעה לhandler המתאים של האינטרפט.

כאשר קורה אינטרפט, המעבד משתמש במזהה של האינטרפט, מחפש בטבלת האינטרפטים, ומעביר את השליטה לhandler המתאים. כלומר:

Hardware -> Interrupt Signal -> CPU -> Interrupt Table -> Interrupt Handler -> Kernel

ב. טבלת syscall היא סוג של מבנה נתונים שמערכת ההפעלה משתמשת בו בשביל לנהל system calls.

כל כניסה בטבלה מתאימה לsyscall ספציפי ומצביעה על handler המתאים שלה, והוא זה שמטפל בsyscall system call.

Application -> syscall signal -> syscall Table -> syscall handler -> Kernel

ג. DMA הוא רכיב שמטרתו להוריד עומס מהCPU. הוא עושה את זה על ידי כך שהוא שולט במעבר של המידע בין הזיכרון הראשי לבין io device. כלומר CPU מעורב רק בהתחלה בקריאה של ההוראה ובסוף התהליך, מעבר לכך השאר מתנהל על ידי DMA.

היתרון העיקרי שלו הוא היעילות בהעברת המידע בין הזיכרון להתקנים שונים, מה שמפנה את CPU להתעסק בדברים אחרים.

חסרון מסוים הוא התחרות על המשאבים, אם לדוגמה יש עכשיו תוכנית בעדיפות גבוהה שדורשת הרבה גישות לזיכרון, אם נעביר במקביל מידע דרך DMA, מצב זה יכול לגרום לעיכובים במערכת ולהוריד את הביצועים שלה.

## שאלה 3

סוג	שימוש	דוגמה	יתרונות	חסרונות
Embedded	משמשות למכשירים עם פונקציונליות ספציפית, בדרכ במכשירים ביתיים, IoT ועוד	FreeRTOS	1. יעילות תוכנית מעצם כך שהותאמו במיוחד לחומרה עליה רצים פונקציונליות מותאמת למשימה לה נועדו	1. יש להן ממשק משתמש מינימלי שמגביל שימוש. 2. קשה לשמר ולעדכן לאור ההתאמה לחומרה
Real-time	משמשות למערכות שצריכות תזמון מדויק ואמינות כמו בתעשייה ביטחונית ומכשירים רפואיים	VxWorks	1. מהירות 2. אמינות, משומשות למערכות קריטיות	1. מסובכות 2. משומשות למשימות ספציפיות ולא גמישות
Servers	משמש בשביל ניהול חומרת שרת ונתינת שירות למספר רב של משתמשים או התקנים ברשת.	Windows Server	1. יודעות לעבוד ע מספר רב של משתמשים בתעבורה גבוהה 2. דר"כ בעלות אבטחה מוגברת לאור סוג המידע שעובר בהן	1. יקרות 2. מסובכות לניהול ושימוש
Batch	משמש לטיפול בכמויות גדולות של משימות דומות. שלא צריכות ממשק משתמש. לדוגמה בחישובים מדעיים	IBM Job Control	1. יעילות, יודעת לעבד קבוצה גדולה של משימות דומות 2. מערכות עם אוטומציה מרובה מה שלא מצריך אינטרקציה עם המערכת	1. חוסר האינטרקציה עם המערכת מייצר הגבלה על משימות שדורשות ממשק עם המשתמש 2. זמני המתנה שיכולים להיווצר מכך שהמשימות מעובדות בקבוצות ולא כיחידים.

## סוני 5

א. חומרה והתקנים –

CPU: AMD Ryzen Zen 2-based processor (8 cores, 16 threads)  
GPU: Custom AMD Radeon RDNA 2-based graphics engine  
RAM: 16 GB GDDR6  
Storage: Custom 825 GB SSD  
HDMI 2.1 output for video  
USB Type-A and Type-C ports  
Ethernet port  
Wi-Fi 6  
Bluetooth 5.1  
DualSense wireless controller  
D audio output via dedicated audio hardware3

ב. מערכת הפעלה –

המערכת רצה על בסיס OrbisOS שהיא מערכת הפעלה שמיועדת לקונסולות.

ג. פרטים על מערכת ההפעלה –

מערכת ההפעלה OrbisOS מבוססת על מערכת FreeBSD והותאמה לקונסולות גיימינג. המערכת מותאמת לאתחול מהיר וטעינה מהירה שהם קריטיים לקונסולות גיימינג וחווית משחק טובה. המערכת מנהלת את המשאבים שלה ונותנת עדיפות לביצועים של המשחקים וממזערת השהיות. המערכת מאובטחת בכדי למנוע פיראטיות וגישה בלתי מורשת לקוד פנימי\חומרה. בעלת ממשק משתמש למשחקים, מדיה ועוד. מותאמת למשחקים מקוונים במהירות גבוהה. תומכת בהרצת משחקים מדורות קודמים של פלייסטיישן. המערכת מאפשרת למשתמש הורדה ועדכונים ברקע מבלי להפריע לזמן המשחק. למערכת אינטגרציה עם הרשת הפלייסטיישן מה שמאפשר לה גישה לשירותים של פלייסטיישן.

## טסלה

לטסלה 2 מערכות הפעלה משמעותיות. אחת מהן היא גם העיקרית, משמשת עבור מערכת הבידור המרכזית ברכב, המערכת הזו מבוססת לינוקס ומותאמת אישית לרכב. המערכת השנייה היא מערכת Real Time שמיועדת לבטיחות של הרכב ונועדה לשירות Autopilot של הרכב. טסלה משתמש במערכות האלה ממספר סיבות-

1. עבור המערכת מבוססת לינוקס-

א. לינוקס מאפשרת גמישות בהתאמה אישית לצרכים ספציפיים של חומרה ותוכנה.

ב. לינוקס היא מערכת קוד פתוח, דבר זה מאפשר לחברת טסלה לשנות את מערכת ההפעלה ללא הגבלות על הרישוי, מה שמפחית לה עלויות ומשפר את השליטה על פיתוח התוכנה.

ג. לינוקס ידועה בתור מערכת הפעלה יציבה ואמינה, על כן השימוש שלה רלוונטי בתעשיית הרכב בה מעט מסובך יותר להוציא שיפורים, תיקונים ועדכונים.

2. עבור מערכת ה-Real Time

א. המערכת מבטיחה זמני תגובה צפויים ומהירים עבור אירועים מידיים, דבר חשוב במקרים של תאונות.

ב. מערכות Real Time מתוכננות להיות אמינות ובטוחות בהקשר של עמידה בסטנדרטים של תעשיית הרכב.

#### שאלה 4

טבלת זמני הרצה (במילי שניות) -

Iterations\Processes	1	2	4	8	16	32	128	256
1	53	54	63	89.375	120.5	114.84	132.47	131.82
2	54	55	63.5	93.125	121.562	113.4	133.04	136.47
100	53	54.5	62.75	98.75	114.25	111.62	130.156	135.23
500	56	55.5	64.25	87.875	112.437	122.46	131.828	136.78
1000	52	57.5	63.25	87.625	119.187	123.812	135.58	135.65
5000	52	55.5	65	86.25	115.75	113.281	130.71	133.32
10000	53	55.5	63.5	85.75	123.25	115.84	130.39	133.511

ניתן לראות מספר דברים עיקריים –

1. כאשר מספר התהליכים גדל, כך זמן הסיום הממוצע גדל.

א. במספר גבוה של תהליכים (16 ומעלה) כל תהליך מתחרה על משאבים של המחשב, התחרות הזאת כנראה מגדילה את זמני הביצוע, יכול להיות שהמחשב עסוק יותר בcontext switch וניהול processes מאשר בחישובים בפועל.

ב. נראה שיש לנו נקודה (8 תהליכים) שבה כמות התהליכים לא משפיעים בצורה טובה על המערכת. כלומר מ-8 ומטה ההבדלים בין הזמנים וכמות התהליכים לא נורא משמעותיים, מעל 8 תהליכים נראה שיש קפיצה משמעותית בזמני הסיום של התוכנית.

2. כאשר מספר האיטרציות גדל, לא נראה שיש השפעה משמעותית על הזמנים הממוצעים, יכול להיות שההשפעה מינורית בהקשר לחומרה עליה רץ הקוד, יכול להיות שהמערכת מבצעת אופטימיזציה לתוכנית אם היא יודעת שהיא הולכת לרוץ מספר רב של פעמים.