בחינה בעקרונות מערכות הפעלה

קראו בעיון לפני שתתחילו בפתרון הבחינה!

1. המבחן מורכב משלושה חלקים.
2. בחלקים א' ו ב' מופיעות שאלות פתוחות. ענו **תשובות מלאות**, **בכתב קריא** ו**בקיצור נמרץ**. אין חובה להשתמש בכל השורות המוקצות לצורך התשובות, אך אין לחרוג מהמקום המוקצה.
3. בחלק ג' (שאלות אמריקאיות) עליכם לבחור בכל פעם בתשובה יחידה מבין התשובות המוצעות ולהקיף בעיגול את אות התשובה שבחרתם.
4. כל חומר עזר אסור לשימוש בזמן הבחינה פרט למחשבון פשוט
5. משך הבחינה - שלוש שעות.

**את התשובות לכל השאלות יש לכתוב בשאלון הבחינה.**

בהצלחה!

## חלק א (55 נקודות)

ענו על **שלוש** השאלות הבאות.

### שאלה 1 (24 נקודות)

כזכור, ספריית Hoard memory allocator עושה שימוש במספר ערמות (heaps). ענו על השאלות הבאות:

|  |  |  |
| --- | --- | --- |
| (6 נק') | א. | לאיזו השפעה הייתם מצפים אם Hoard memory allocator היה עושה שימוש ב 2 ערמות פר מעבד במקום ערמה אחת? אפשר להניח שפונקציית ערבול שבוחרת את הערמה היא:  heapNumber = threadID modulo 2 \* P  כאשר P הוא מספר המעבדים.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (6 נק') | ב. | Hoard memory allocator מנהל בלוקים פנויים בתוך ב superblock בעזרת המדיניות LIFO. הסבירו כיצד הדבר עוזר בשיפור ה locality של האיזורים המוקצים.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |  |
| (6 נק') | ג. | רשמו את הנוסחה המחשבת את תקורתה של ההקצאה בתוך הסופר-בלוק. תתיחסו לסופר-בלוק שלא נותר בו מקום להקצאה. כמובן שהשאלה מתייחסת לגודל ההקצאה המבוקשת שהיא קטנה או שווה ל S/2, כאשר S הוא גודל הסופר-בלוק. הנוסחה היא הפונקציה של הפרמטרים הבאים: גודל השטח המבוקש המעוגל לחזקת 2 כלפי מעלה וגודל ה header למימוש של הרשימה המשורשת.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (6 נק') | ד. | היכן מקצה Hoard memory allocator את שטחי הזיכרון שגודלם יותר מ S/2, כאשר S הוא גודל הסופר-בלוק? השלימו את הציור של מפת הזיכרון של תהליך בלינוקס והצביעו על מקום ההקצאות של Hoard memory allocator על גבי המפה. ציינו מהו תפקידו של כל סגמנט במפה.  0xffffffff  1 GB  3 GB |

0x00000000

### שאלה 2 (16 נקודות)

הוכיחו או הפריכו את הטענות הבאות לגבי אלגוריתם הבנקאי (banker algorithm).

|  |  |  |
| --- | --- | --- |
| (5 נק') | א. | אם אלגוריתם הבנקאי גילה שהמערכת נמצאת במצב לא בטוח, אז בהכרח שהמערכת תגיע לקיפאון בעתיד.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (5 נק') | ב. | אם אלגוריתם הבנקאי גילה שהמערכת נמצאת במצב בטוח, אנחנו יודעים בוודאות שלא יתכן קיפאון בעתיד בהשתתפות המשאבים עליהם הורץ האלגוריתם.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (6 נק') | ג. | מהן שתי המגבלות העקריות של אלגוריתם הבנקאי אשר הופכות אותו לבלתי שימושי באופן מעשי? הניחו שמדובר על תהליכים שתופסים משאבים לאורך תקופה ארוכה (תקופה של שעות או של ימים).  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### שאלה 3 (15 נקודות)

במערכת מקבילית ישנם T תהליכונים (threads), כאשר T הוא מספר קבוע קטן מ 1024. במערכת מספר מעבדים. בכל עת התהליכונים יכולים להיות מועברים למעבד אחר (לא חשוב כעת לפי איזה קריטריון). הבעיה היא שהתהליכונים יכולים "ליפול" בזמן לא ידוע ומסיבה לא ידועה. כאשר תהליכון "נופל" הוא מפסיק לבצע את ההוראות שלו ולא חוזר לעולם.

תהליכונים יכולים לבצע פעולות שדורשות מתהליכונים אחרים להמתין. הפעולות הללו מוגבלות בזמן וחסומות ע"י הזמן שלוקח לבצע את הקוד הבא:

|  |
| --- |
| int counter;  for (i=0; i++; i<MAX\_INT) {  counter++;  } |

נרצה לנצל את העובדה הזאת כדי לזהות נפילת תהליכונים במערכת המקבילית, הרי אם תהליכון כלשהו מחזיק במנעול לאורך הזמן שעולה על זמן הביצוע של הלולאה דלעיל – הרי שזה אומר שהתהליכון "נפל".

הוצעו 2 הצעות לפתור את הבעיה בעזרת מנעול אשר נפתח בעצמו אם הוא תפוס יותר מדי זמן.

ההצעה הראשונה הייתה כדלהלן:

|  |  |  |
| --- | --- | --- |
| typedef struct Lock\_t {  bool b;  } Lock; | void down(Lock\* lock) {  int counter = 0;  while ( !t&s( &( lock 🡪b) ) ) {  counter++;  if (counter == MAX\_INT) {  up(lock);  }  } *// end of while*  } | void down(Lock\* lock) {  bzero(lock🡪b);  } |

ההצעה השניה הייתה:

|  |  |  |
| --- | --- | --- |
| typedef struct Lock\_t {  bool b;  int counter;  } Lock; | void down(Lock\* lock) {  int oldCounter;  while ( !t&s( &(lock🡪b) ) ) {  oldCounter =  fetch&Increment( &(lock🡪counter) );  if ( (oldCounter +1) == MAX\_INT) {  up(lock);  }  } *// end of while*  lock🡪 counter = 0;  } | void up(Lock\* lock)  {  lock🡪b = 0;  lock🡪counter = 0;  } |

הבהרות:

* הניחו ש-t&s() היא פקודה אטומית אשר קוראת ערך מכתובת ואם הוא אפס משנה אותו באופן אטומי ל-1. הפקודה מחזירה true אם הצליחה לשנות ערך. אחרת היא מחזירה false.
* הניחו ש- fetch&Increment() היא פקודה אטומית אשר קוראת ערך מכתובת, מקדמת את הערך ב-1 ואחרי כן כותבת אותו בחזרה וכל זה באופן אטומי. הפקודה מחזירה את הערך הישן שנקרא (לפני עדכון).

|  |  |  |
| --- | --- | --- |
| (7 נק') | א. | הסבירו מדוע ההצעה הראשונה יכולה לגרום לכך שלא תתקיים מניעה הדדית (mutual exclusion) |

הסבר:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| (8 נק') | ב. | הוכיחו או הפריכו:  ההצעה השניה גם כן סובלת מבעיה של חוסר מניעה הדדית. |

הסבר:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## חלק ב (25 נקודות)

ענו על **חמש** השאלות הבאות. משקל כל שאלה 5 נקודות.

### שאלה 4

הסבירו מה עושה התוכנית הבאה:

|  |
| --- |
| #define handle\_error(msg) \  do { perror(msg); exit(EXIT\_FAILURE); } while (0)  int  main(int argc, char \*argv[])  {  char \*addr;  int fd;  struct stat sb;  off\_t offset, pa\_offset;  size\_t length;  ssize\_t s;  if (argc < 3 || argc > 4) {  fprintf(stderr, "%s file offset [length]\n", argv[0]);  exit(EXIT\_FAILURE);  }  fd = open(argv[1], O\_RDONLY);  if (fd == -1)  handle\_error("open");  if (fstat(fd, &sb) == -1) /\* To obtain file size \*/  handle\_error("fstat");  offset = atoi(argv[2]);  pa\_offset = offset & ~(sysconf(\_SC\_PAGE\_SIZE) - 1);  /\* offset for mmap() must be page aligned \*/  if (offset >= sb.st\_size) {  fprintf(stderr, "offset is past end of file\n");  exit(EXIT\_FAILURE);  }  if (argc == 4) {  length = atoi(argv[3]);  if (offset + length > sb.st\_size)  length = sb.st\_size - offset;  /\* Can't display bytes past end of file \*/  } else { /\* No length arg ==> display to end of file \*/  length = sb.st\_size - offset;  }  addr = mmap(NULL, length + offset - pa\_offset, PROT\_READ,  MAP\_PRIVATE, fd, pa\_offset);  if (addr == MAP\_FAILED)  handle\_error("mmap");  s = write(STDOUT\_FILENO, addr + offset - pa\_offset, length);  if (s != length) {  if (s == -1)  handle\_error("write");  fprintf(stderr, "partial write");  exit(EXIT\_FAILURE);  }  exit(EXIT\_SUCCESS);  } |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### שאלה 5

מהו ההבדל בין TRAP instruction לבין פסיקת חומרה (hardware interrupt)?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### שאלה 6

שרטטו את ה layout של מערכת הקבצים ext2. כלומר, כיצד היא מחלקת את הבלוקים בדיסק ל groups, מה ממוקם בכל group והיכן?

שרטוט:

### שאלה 7

נתון מנגנון תזמון תהליכים בשיטת ההגרלה (lottery scheduling). כיצד, בלי לשנות קוד של מתזמן, ניתן לקבל מהמנגנון הזה תזמון תהליכים שייתן לתהליכים המתוזמנים זמני תגובה כמו round robin?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### שאלה 8

השלימו את השרטט. כיצד מתבצע תרגום כתובת לוגית לכתובת פיזית באמצעות page table?

שרטוט:



## חלק ג (20 נקודות)

ענו על כל **ארבע** שאלות רב-הברירה (אמריקאיות) הבאות. משקל כל שאלה 5 נקודות.

בכל שאלה יש לבחור את התשובה הנכונה ולהקיף בעיגול את אות התשובה שבחרתם.

### שאלה 9

מתי הכרחי לחסום סיגנלים (signals)? תזכורת: חסימת סיגנלים מתבצעת על ידי עדכון של מסכת סיגנלים חסומים (blocked signals mask) והעברתה ל sigaction.

1. כאשר תהליך עומד לגשת למבנה נתונים שגם אחד או יותר מה-signal handlers בתוכנית ניגשים אליו.
2. כאשר תהליך שמטרתו לפגוע במערכת (וירוס, למשל) רוצה למנוע מתהליכים אחרים לשלוח לו סיגנלים שיגרמו למותו (כגון SIGKILL).
3. כאשר תהליך נמצא בשליטת debugger ורוצה להשתחרר ממנו על ידי חסימת סיגנל SIGSTOP
4. כאשר תהליך נמצא בכניסה לשגרת טיפול בפסיקה.

### שאלה 10

איזו מבין התופעות הבאות איננה קשורה בהכרח למקביליות בין הליכים/תהליכונים?

א. קיפאון (deadlock)

ב. בעיית הקטע הקריטי

ג. סחרור בזיכרון (threshing)

ד. בעיית המרוץ (race condition)

### שאלה 11

כאשר מדובר במבנה מערכת הפעלה לפי מודל שרת-לקוח (client-server model), מהי התכונה אשר מהווה חיסרון מובהק של המודל?

1. העדר מבנה כלשהו. המערכת היא אוסף שגרות אשר כל אחת מהן יכולה לקרוא לשגרה אחרת מן האוסף.
2. חוסר אפשרות התאמה למערכות מבוזרות (distributed systems).
3. התקורה (overhead) שבתקשורת בין רכיבי המערכת.
4. חוסר אפשרות של תקשורת מאובטחת

### שאלה 12

להלן תוצאות הבדיקה של עקביות מערכת הקבצים:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Block number | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| Blocks in use | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| Free blocks | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |

מה אפשר להסיק לפי הנתונים הנ"ל?

1. מערכת הקבצים הנה עקבית.
2. מערכת הקבצים איננה עקבית – בלוקים חסרים.
3. מערכת הקבצים איננה עקבית – בלוקים משוכפלים.

מערכת הקבצים אינה עקבית – בלוקים משובשים.

**בהצלחה!**