הצעה למבחן במערכות הפעלה על בסיס הפרמטרים שעמית הגדיר

(אם נבחר 3 שאלות פתוחות כ"א 20 נקודות אחת למועד א אחת למועד ב ואחת תפתח לסטודנטים מבחן לדוגמא) אני מצרף 9 שאלות פתוחות.

אמריקאיות. – אני מניח שיהיו 8 שאלות במבחן ולכן אני מצרף 24 שאלות

1. במערכת קבצים מסוג UFS ניתן להגדיל קבצים קיימים אבל לא ניתן לייצר קבצים חדשים מה יכולה להיות הסיבה
   a. נגמרו הINODES
   b. נגמרו הBLOCKים
   c. נגמר המקום ב root דירקטורי
   d. נגמר המקום בסופרבלוק
   e. נגמרו הFRAGMENTS
2. איזה מהפונקציות הבאות לא חוסמת
   a. listen(2)
   b. accept(2)
   c. scanf(3)
   d. recv(2)
   e. כולן חוסמות
3. מי מהבאים זמין לנו בקרנל
   a. משתנים מטיפוס DOUBLE
   b. פונקציית printf(3)
   c. System call write(2)
   d. פונקציות ספריה לרבות הספריה הסטדנרטית
   e. SHELL להרצת פקודות בהרשאת ROOT
4. איזה מהפרויקטים הבאים ניתן לצפות שיהיה משותף למספר הפצות של linux
   a. כל התשובות נכונות
   b. קרנל
   c. קומפילר gcc
   d. שרת ווב apache
   e. דפדפן firefox או iceweasel
5. איזה מהבאים זמין בchar device ואיננו זמין בblock device
   a. אף תשובה איננה נכונה
   b. read
   c. write
   d. mmap
   e. lseek
6. נדרש לנעילה על מנת לממש את תבנית העיצוב הבאה
   a. Singleton
   b. Factory
   c. Decorator
   d. Fascade
   e. Reactor

7. איזה מהבאים לא מיוצג בלינוקס בעזרת קובץ
   a. POSIX MUTEX
   b. Unix domain socket
   c. TCP socket
   d. Data gram sock
   e. Char device
8. רישיון GPL הופק על ידי ארגון שבראשו עמד
   a. Richard Stallman
   b. Dennis Ritchie
   c. Linus Torvalds
   d. Eric S. Raymond
   e. Doug Lea
9. DOUG LEA כתב את המערכת החשובה הבאה
   a. ניהול הHEAP (malloc ו free)
   b. ניהול מערכת הקבצים
   c. ספר הדיזין פטרנז POSA
   d. שפת C
   e. היוניקס הראשון במעבדות BELL LABS
10. בתוכנית מרובת THREADS אחד הטרדים קרא ל(exit(2
   a. כל הטרדים יסתיימו
   b. הטרדים ימשיכו לרוץ
   c. הסטנדרט לא מגדיר. כל ישום יעשה מה שיבחר.
   d. תלוי אם קימפלנו עם דגל -pthread
   e. נלך לבצע את הsignal handler של SIGCHLD
11. הסבר מה זה ADVISORY LOCKING
   a. נעילה שבא אם תהליך ינסה לשאול האם המשאב נעול יקבל תשובה אבל אם פשוט ינסה לגשת יתאפשר לו לגשת גם אם המשאב נעול.
   b. נעילת משתמש על תהליך או קובץ.
   c. נעילת מערכת הפעלה על תהליך או קובץ.
   d. נעילה שבה אנחנו ממליצים על פעילות אחרת אם המשאב נעול
   e. נעילה שבה אנחנו מדווחים מתי נשחרר את המשאב אם המשאב נעול
12. באיזה תנאים שימוש בפרגמנטים לא ישפר את ביצועי מערכת קבצים
   a. אם מערכת הקבצים נמצאת על SSD
   b. אם אנחנו תומכים רק ב3 רמות של פרגמנטים
   c. אם מערכת הקבצים נמצאת על דיסק מגנטי
   d. אם מערכת הקבצים נמצאת על טייפ גיבוי
   e. אם מערכת הקבצים תומכת בהצבעות עקיפות ^3 ועקיפות ^4.
13. ניתן לצפות ש
   a. בעוד כעשור ה api של לינוקס ישאר זהה למצבו היום. לעומת זאת קרנל מודול יתנהגו אחרת.
   b. בעוד כעשור ה api של לינוקס וגם של קרנל מודולז ישאר זהה למצבו היום.
   c. בעוד כעשור יהיו שינויים משמעולים גם ב api של לינוקס וגם של קרנל מודולז
   d. בעוד כעשור של קרנל מודולז ישאר זהה למצבו היום ככול הנראה יהיו הבדלים ניכרים בAPI של לינוקס
   e. לא ניתן לחזות את העתיד

14. האם טרד יכול לקרוא ערך (בעזרת מצביע) הנמצא במחסנית של טרד אחר (אותו תהליך)
   a. כן.
   b. לא.
   c. תלוי בגירסא של לינוקס
   d. תלוי בגירסא של הספריה הסטנדרטית
   e. כן. בקרנל ספייס. לא ביוזר ספייס.
15. תהליך אבא ותהליך ילד חולקים בינהם
   a. בזכרון ששותף (לדוגמא MMAP עם MAP SHARED) לפני הקריאה לFORK.
   b. בכל המשאבים
   c. באף משאב
   d. בקבצים הפתוחים בחם
   e. בinstruction pointer
16. תבנית העיצוב לא קשורה לשימוש בטרדים
   a. Factory
   b. Singleton
   c. Guard
   d. Active object
   e. Pipeline
17. מה החסרון של הגדלת גודל הבלוק במערכת קבצים
   a. בזבוז מקום
   b. יותר פעולות SEEK
   c. פחות פעולות SEEK
   d. ניתן לתמוך בקבצים יותר קטנים
   e. נידרש ליותר הצבעות עקיפות לקבצים
18. כאשר אנחנו משתמשים בSOCK STREAM אנחנו יכולים להיות בטוחים ש
   a. כל מה שהתקבל – אכן נשלח. התקבל תקין ובסדר שהתקבל.
   b. כל מה שהתקבל – אכן נשלח והתקבל תקין. יתכן שנשלח בסדר שונה
   c. כל מה שהתקבל – אכן נשלח התקבל תקין ובסדר שהתקבל. בנוסף מה שהתקבל במספר הוראות RECV נשלח באותו מספר הוראות SEND (פקטים לא נדבקים ולא נשברים)
   d. כל מה שהתקבל – אכן נשלח והתקבל תקין. יתכן שנשלח בסדר שונה. בנוסף מה שהתקבל במספר הוראות RECV נשלח באותו מספר הוראות SEND (פקטים לא נדבקים ולא נשברים)
   e. כל מה שהתקבל – אכן נשלח התקבל תקין ובסדר שהתקבל. בנוסף מה שהתקבל במספר הוראות RECV נשלח באותו מספר הוראות SEND (פקטים לא נדבקים ולא נשברים) בנוסף הצד השני מחובר עדין לאינטרנט (אני אדע מייד (כהרף עין.) אם יתנתק מכל סיבה שהיא)
19. איזה מהפונקציות הבאות משחררת נעילה על MUTEX
   a. Pthread_cond_wait(3)
   b. Pthread_muted_init(3)
   c. Pthread_cond_init(3)
   d. Pthread_mutex_lock(3)
   e. Pthread_mutex_trylock(3)

20. איזה מהפונקציות או קריאות המערכת הבאות חוזרת פעמים
    a. fork(2)
    b. wait(2)
    c. scanf(3)
    d. pthread_create(3)
    e. signal(3)
21. איזה מהפונקציות או קריאות המערכת הבאות יכולה לסייע לנו לדעת מדוע קריאת מערכת נכשלה
    a. perror(3)
    b. printf(3)
    c. ptrace(2)
    d. wait(2)
    e. setsockopt(2)
22. כיצד אנחנו יכולים לדעת מדוע תהליך הסתיים בהנחה שהסתיים בצורה לא תקינה
    a. לתחקר את קובץ ה core dump שהשאיר
    b. האבא יקבל SIGCHLD שכולל את המידע
    c. נקבל זומבי שיאמר לנו מדוע נוצר
    d. יש דימון במערכת המטפל בתהליכים שנכשלו
    e. נוכל לבצע cat /var/log/kern.log
23. נדרשות הרשאות מנהל (לדוגמא sudo)
    a. על מנת להסיר קרנל מודול
    b. על מנת למחוק קובץ שלי
    c. על מנת לקמפל קרנל מודל
    d. כדי להרוג תהליך שלי
    e. על מנת לבצע bind(2) על SOCKET
24. לתהליך
    a. יש גישה לזכרון שלו בלבד
    b. יש גישה לזכרון שלו ושל ילדיו
    c. יש גישה לזכרון שלו ושל כל התהליכים שנוצרו על ידי אותו משתמש
    d. יש גישה לזכרון שלו ולהתקני החומרה שהוא משתמש בהם
    e. יש גישה לזכרון שלו ושל ילדיו פרט להתקני החומרה שהם משתמשים בהם

<u>שאלות ניתוח מערכת</u>

1. חברת אפל שיחררה לעולם מספר רב של גירסאות של מערכת הפעלה מקינטוש בין השנים 1984 ל1999. בשנת 1999 אפל החליטה "לעבור דור" במערכת ההפעלה שלה ושקלה מספר אפשרויות. (בין היתר שקלה גם לרכוש את חברת BE ולבסס את מערכת ההפעלה שלה על מערכת BEOS. לבסוף אפל ביצעה בחירה שיחררה את מערכת הפעלה OSX שהתבססה על גרעין דרווין וfreebsd שנכתבו תחת רישיון BSD (רישיון קוד פתוח). אפל שיחררה את הגירסא הראשונה של מערכת OSX במארס 2001.

בשנת 2001 היה בעולם שימוש נרחב בלינוקס (אולי לא כמו היום אבל בהחלט באופן הרבה יותר דומיננטי מדרווין ומFREEBSD) וללא ספק מקבלי ההחלטות באפל הכירו את לינוקס.

הסבר אילו <u>שיקולי רישיון</u> יכלו להיות לאפל כשבחרו לא להשתמש בלינוקס אלא בBSD כגרעין למערכת ההפעלה שלהם.

הבהרה – אתם נדרשים להתייחס לנושאי רישוי בלבד. יש בין המערכות גם הבדלים טכנים וללא ספק גם הם היוו משקל אבל שאלה זאת מתיחסת לרישוי קוד פתוח בלבד.

2. TRUSTZONE הוא מוד אבטחה במעבד ARM המאפשר להריץ מערכת הפעלה נוספת לצרכי אבטחה (בנוסף למערכת ההפעלה המרכזית). לדוגמא על פלאפון יכולה לרוץ מערכת הפעלה ANDROID המריצה ישומים רבים ומערכת הפעלה נוספת כגון 4OKL המעניקה שרותי אבטחה בלבד (לדוגמא הזדהות חכמה ושמירה על מידע אישי) המתקשרת עם אפליקציה בנקאית. על סמך הנושא שנלמד בקורס בקרנל האם נעדיף שמערכת ההפעלה המאובטחת (לדוגמא 4OKL) תהיה מערכת מיקרו קרנל, מונוליטיק קרנל או היברידית. כיצד היתרונות והחסרונות של הבחירה יבואו לידי ביטוי

שים לב – מערכת ההפעלה המאובטחת מיועדת בעיקר לביצוע מספר מוגבל של פעולות אבטחה ולרוץ זמן קצר יחסית (כרשר רוב הזמן מעבד יוקדש מין הסתם למערכת ההפעלה העשירה ANDROID מערכת ההפעלה המאובטחת תבצע שרותים בודדים עבור מספר קטן של ישומים)

3. בשרת DVR אוספים נתונים ממספר מצלמות אבטחה (לדוגמא 10 מצלמות המקיפות מבנה) ושומרים את הסרטונים השמורים לתקופה. (שלאחריה הם נמחקים. לרוב בלי שצפו בהם אפילו פעם אחת) בשרת נאגרים סרטונים בזכרון לתקופה של מספר שניות ונכתבים לדיסק מספר מגה בייטים בכל פעם. דני מציע לשדרג את השרת ולעבור מדיסקים מגנטים לדיסקים בטכנולוגיות SSD - SOLID STATE. הביעי את דעתך על הצעתו של דני.

אם הנתונים נאספים לא ממצלמות אבטחה אלא ממספר VLOGERים (כלומר VIDEO BLOGGER) המוצגים למספר רב של משתמשים (כמו בYOUTUBE) – VLOGGER ממוצע מקליט מספר סרטונים בני כמה דקות ביום. (סה"כ מנוהלים עשרות אלפי סרטונים)

הסרטונים נצפים אחר כך אלפי פעמים ביום על ידי מספר רב של אנשים כל הזמן.

<u>שאלות תכנון</u>

4. תכנן שרת דוא"ל עבור ספק שרות אינטרנטי (כמו יאהו מייל או GMAIL). השרת צפוי לקבל דוא"ל (כולל מכתבים רגישים) עבור משתמשים חינמים ומשתמשים משלמים. צפי הלקוחות של החברה הוא מיליון משתמשים בשנה הראשונה כל אחד מקבל בממוצע 10 הודעות דו"אל ביום ומתחבר מספר פעמים ביום על מנת לבדוק הודעות.
השרת צריך לתמוך בקריאת ההודעות על גבי אתר ווב (כמו גוגל) שליחת הודעות בפרוטוקול SMTP (כפי שלמדתם בקורס תקשורת) וקריאת הודעות על ידי תוכנה (לדוגמא בפרוטוקול 3POP)

אפשר להניח שמעבר לתמיכה בפרוטוקול 3POP ו smtp לא נדרשת תמיכה בפרוטוקולים נוספים. יש להתיחס לשיקולים כמו ביצועים (כמה זמן לוקח למערכת לטפל בקריאת דוא"ל או שליחת דוא"ל) וגידול (מה יקרה אם יהיו 10 מיליון משתמשים שמקבלים 40 הודעות ביום), אבטחת מידע. זמינות. וכדומה.
חובה וחשוב לציין כל ישות שמערכת ההפעלה רואה (כלומר קרנל מודול, תהליך, קובץ וכדומה) בקשר לישויות ברזולוציה נמוכה יותר (לדוגמא ישומים בתהליך תוכן בקובץ וכדומה) תוכלו לציין אותם אם אתם רואים בהם חשיבות אבל אין חובה לציין אותם.

5. חברת DILIGENT נקנתה על ידי IBM בשנת 2008. החברה יצרה מוצרי VTL – כלומר דימתה טייפים של גיבוי בעזרת דיסקים שאפשרו לכל מוצרי הגיבוי שעבדו עם טייפים אמיתים לעבוד.

בהנתן לינוקס שיש לו מספר אפליקציות היודעות לעבוד עם טייפ גיבוי (התקן חומרה) וספריות טייפים כאלה. עליכם לתכנן דרייבר לטייפ וירטואלי השומר את הנתונים על גבי דיסק קשיח. (במקום על ידי טייפ פיזי) התייחס לפרמטרים כמו ביצועים עמידות המערכת ותאימות אחורה.
חובה וחשוב לציין כל ישות שמערכת ההפעלה רואה (כלומר קרנל מודול, תהליך, קובץ וכדומה) בקשר לישויות ברזולוציה נמוכה יותר (לדוגמא ישומים בתהליך תוכן בקובץ וכדומה) תוכלו לציין אותם אם אתם רואים בהם חשיבות אבל אין חובה לציין אותם.

6. חברת OKCUPID מייצרת אתר אינטרנט המאפשר לרווקים ולרווקות ישראלים להכיר. החברה קולטת משתמש.ת חדש.ה קולטת מידע על המשתמש.ת והעדפותיו.ה ולאחר מכן מציעה לו.ה MATCHים המתאימים לו.ה. החברה מאפשרת לשלוח הודעות בין מנויים ביצוע שיחות וכדומה.

החברה מעוניינת לפתח מודול חדש של משחקי חברה. במסגרת זו ישחקו מספר מנויים במשחק (לדוגמא "מונופול", "הרמז" או "ארץ-עיר"). תכננו את מודול משחקי החברה.

הערות – תוכלו להניח כל הנחה שתרצו בה לגבי השרות (לדוגמא מספר המשתמשים או המשחק – בתנאי שמדובר במשחק חברה המאפשר למשתתפים לייצר אינטרקציה. משחקים כמו "שחמט" לא מתאימים) תוכלו להניח קיום טכנולוגיות מודרניות (לדוגמא שיחת וידאו) הנחות

a. לקוחות החברה נשמרים בבסיס נתונים (ניתן להניח – קובץ)
b. התקשורת עם כל לקוחות החברה הינה בעזרת TCP על גבי IP בלבד.
c. במידת האפשר כל משתתף.ת יופנה למשחק עם שחקנים המתאימים לו לפי העדפותיו.ה
d. השחקנים יכולים לראות את ה"פרופיל" של השחקנים שמשחקים איתם במהלך המשחק.
e. לחברה יש בסיס לקוחות של מאות אלפי מנויים.
יש הרבה משחקים והרבה שחקנים בו זמנית. (ניתן להניח שבשעות השיא מחוברים לשרתי החברה כמה אלפי רווקים ומתרחשים עשרות או מאות משחקים)

חובה וחשוב לציין כל ישות שמערכת ההפעלה רואה (כלומר קרנל מודול, תהליך, קובץ וכדומה) בקשר לישויות ברזולוציה נמוכה יותר (לדוגמא ישומים בתהליך תוכן בקובץ וכדומה) תוכלו לציין אותם אם אתם רואים בהם חשיבות אבל אין חובה לציין אותם.

באגים בקוד

7. לפניכם חלק קוד של שרת. בקוד יש באג. מצא אותו ותקן אותו. יש לציין את כל השורות שיש לערוך בהם שינוי

```
int main(void) {
   struct stat statbuf;
   int fp = open("/some/shared/resource/file", "r");
   fstat(fp, &statbuf);

   int *shared = mmap(
      /* addr= */ NULL,
      /* length= */ statbuf.st_size - 1,
      /* prot= */ PROT_READ | PROT_WRITE,
      /* flags= */ MAP_ANONYMOUS,
      /* fd= */ fp,
      /* offset= */ 0);

   pid_t child;
   if ((child = fork()) == 0) {
      // process forked. The child can access the 'shared'
      // object and perform read and write operations on it.
     // the memory is shared
   }
      // process forked. The parent can access the 'shared'
      // object and perform read and write operations on it.
     // The memory is shared
```

8. לפניכם חלק קוד של שרת. שתי הפונקציות המתועדות רצות במקביל בטרדים שונים. בקוד יש באג. מצא אותו ותקן אותו. יש לציין את כל השורות שיש לערוך בהם שינוי

```
void* enable_logger(void *args)
{
...
pthread_mutex_lock(&log_m);
pthread_mutex_lock(&buf_index_m);
for (int i = 0; i <= logger.index; ++i)
{
if (logger.buffer[i] != (char*) NULL)
{
const size_t log_lvl_str_index = (size_t)logger.
log_levels[i] - 1;
printf(''LOG:%s:%s\n'', log_lvl_str[log_lvl_str_index], logger.buffer[i]);
free(logger.buffer[i]);
logger.buffer[i] = (char*)NULL;
}
}
logger.index = -1;
pthread_mutex_unlock(&buf_index_m);
pthread_mutex_unlock(&log_m);
...
}

void add_log(const char *log, enum LogLevel level)
{
char *llog = (char*)malloc(sizeof(char)*strlen(log));
strcpy(llog, log);
if (logger.index >= BUFFER_SIZE || log_level > level) return;
++logger.index;
logger.buffer[logger.index % BUFFER_SIZE] = llog;
logger.log_levels[logger.index % BUFFER_SIZE] = level;
logger.index = logger.index % BUFFER_SIZE;
}
```

9. לפניכם חלק קוד של שרת. בקוד יש באג. מצא אותו ותקן אותו. יש לציין את כל השורות שיש לערוך בהם שינוי

```
fd_set master;
int fdmax, listener;
…
FDZERO(&master)
…
FD_SET(listener, &master);
fdmax=listener;
for(;;) {
    if (select(fdmax+1, &master, NULL, NULL, NULL) == -1) {
        perror("select");
        exit(4);
    }

    // run through the existing connections looking for data to read
    for(i = 0; i <= fdmax; i++) {
        if (FD_ISSET(i, &master)) { // we got one!!
            if (i == listener) {
                // handle new connections
                addrlen = sizeof remoteaddr;
                                newfd = accept(listener,
                                        (struct sockaddr *)&remoteaddr,
                                        &addrlen);

                                if (newfd == -1) {
                    perror("accept");
                } else {
                    FD_SET(newfd, &master); // add to master set
```

```
                if (newfd > fdmax) {    // keep track of the max
                    fdmax = newfd;
                }
                printf("selectserver: new connection from %s on "
                    "socket %d\n",
                                                    inet_ntop(remoteaddr.ss_family,
                                                        get_in_addr((struct
sockaddr*)&remoteaddr),
                                                        remoteIP, INET6_ADDRSTRLEN),
                                                    newfd);
            }
        } else {
            // handle data from a client
            if ((nbytes = recv(i, buf, sizeof buf, 0)) <= 0) {
                // got error or connection closed by client
                if (nbytes == 0) {
                    // connection closed
                    printf("selectserver: socket %d hung up\n", i);
                } else {
                    perror("recv");
                }
                close(i); // bye!
                FD_CLR(i, &master); // remove from master set
            } else {
                // we got some data from a client
                for(j = 0; j <= fdmax; j++) {
                    // send to everyone!
                    if (FD_ISSET(j, &master)) {
                        // except the listener and ourselves
```

```
                    if (j != listener && j != i) {
                        if (send(j, buf, nbytes, 0) == -1) {
                            perror("send");
                        }
                    }
                }
            }
        }
    } // END handle data from client
  } // END got new incoming connection
 } // END looping through file descriptors
} // END for(;;)--and you thought it would never end!
```