**מעבדה במערכות הפעלה :מפגש 1**

* **הנחות על ידע מוקדם** 
  + פקודות בסיסיות
    - ls / cd / cp / mv / rm / mkdir
    - <https://kinsta.com/blog/linux-commands/#the-mostused-linux-commands>
  + pipe/more/grep
    - שרשור פלט לקלט של פקודה הבאב
    - חיפוש בפלט
  + UP/DOWN (הרצת פקודה אחרונה)
  + הרצה של תוכנית ברקע (&)
    - bg/fg - הפיכת תהליך אחרון מהרצה ברקע להרצה בחזית (ולהפך)
  + עורך: eclipse/VI/PICO
  + FTP

**החומרים עבור המפגש:**

* + דף עבודה (לפני כל מפגש יפורסם דף משימות )
  + תיקית תוכניות: lab1\_2\_prog

**הגשה**

* + הגשות בזוגות (יש להגיש רק את הסעיפים המסומנים באדום)
  + הגשה אחת עבור המשימות של השבוע ושל השבוע הבא
  + לגבי דף העבודה הזה: תשובות לכל הסעיפים בשתי המשימות

**לפני שמתחילים לעבוד על דף משימות:** להתחבר לשרת לינוקס של המכללה לפי ההוראות של "הוראות התחברות לשרת לינוקס" (נמצא במודל).

**נושא: כלים ב-LINUX להצגת מידע על תהליכים**

* + **מטרות:**
    - הכרת כלים פשוטים ב-LINUX להצגת מידע על תהליכים (צריכת CPU/זיכרון)
    - הבנת מושג תוכנית bound “ CPU" לעומת “ “ IObound

**תוכנית C (מקור) בהם נשתמש:**

<https://github.com/Bongholio/OS-Labs>

* + - cpu.c/ print\_cont.c (תיקיה: cpu\_io\_bound )
    - memory\_use.c (תיקיה: use\_free )

הערה: ניתן גם להעתיק את התיקיות מ-

margulisdav@cs:~/new\_dir/lab1\_2\_prog

**מטרה משנית (מאוד חשובה):** שימוש ב-MAN PAGES ו-GOOGLE למציאת מידע נחוץ

Hard Disk

(כונן קשיח/זיכרון משני)

Main memory

( זיכרון ראשי)

CPU

(מעבד)

External Devices (I/O, .. )

(התקני קלט/פלט, ...)

* 1. **פקודת PS**
     + ps - report a snapshot of the current processes
     + תשתמשו ב- GOOGLE ללמוד יותר על הפקודה
     + תנסו פקודה: **ps -aux**
       1. איך להציג תהליכים השייכים לך ? תרשמו פקודה
       2. איך להציג תהליכים בשם מסוים ?

**רמז:** תשתמשו ב-grep

* 1. **פקודת TOP** 
     + מה עושה הפקודה ומה ההבדל העיקרוני בין TOP ל-PS ? תסבירו בקצרה
     + מה המשמעות של PID המוצג עבור כל תהליך (גם בפקודה PS) ?
     + איך אפשר לדעת את ה-PID של TOP (**רמז:** צריך לבדוק מ-SHELL אחר)

**משימה 1**

תסתכלו על תוכניות **cpu.c/ print\_cont.c**

* + - 1. תסתכלו על תוכניות : איזה מן התוכניות לדעתכם היא CPU bound ואיזה IO bound ? תסבירו בקצרה
      2. תקמפלו ותריצו את שתי התוכניות (cpu.c/ print\_cont.c) ותבדקו את צריכת זמן ריצה (CPU)
         * לקמפל תשתמשו ב:gcc cpu.c -o cpu.a
         * להריץ ./cpu.a
      3. תריצו כל אחת מהתוכניות בנפרד ותענו על שאלות הבאות
         * מהו PID של כל אחד מ-2 התהליכים המריצים את התוכנית ? תעצרו ותריצו שוב: האם PID השתנה ?
         * מה אחוז ה-CPU הנצרך עבור כל אחד מהתהליכים ?

תשתמשו TOP

* + - * + מה הוא מצב התהליכים כפי שמוצג ע''' שדה’ ‘S בפקודת TOP ?
      1. תריצו את שתי התוכניות ביחד ( תחשבו איך אפשר לעשות את זה)
         * מה הוא מצב של שני התהליכים התהליכים ? **תוסיפו הדפס מסך של הפלט TOP המראה רק את שני התהליכים (רמז: תשתמשו ב-grep המתאים)**

**רמז**: מה הגדרה של המצב בו נמצאים התהליכים

**להגשה**: צילומי מסך של המשמות

* + - 1. על בסיס שתי התוכניות תכתבו תוכנית כך שצריכת ה-CPU (מוצגת עי'' TOP) היא בערך 50% (עיי' ניסוי ותהיה )

**יש להגיש את התוכנית עצמה תוצאות של הדפסת TOP.**

* 1. **פקודת FREE - בדיקת כמות הזיכרון הנצרכת ע''י תהליך**

[**https://www.geeksforgeeks.org/free-command-linux-examples**](https://www.geeksforgeeks.org/free-command-linux-examples)

**מטרה**:  **הבנה מה הוא ההבדל בין זיכרון ראשי וזיכרון SWAP**

* + - **מה עושה פקודת FREE ?** 
      * + מה משמעות ארגומנט ‘-m’ ו-‘-g’?

**הערה:** 1KB = 1024

* + - * + מה משמעות ‘mem’ בפלט של הפקודה ?
    - כמה זיכרון פיסי ס"כ יש במערכת שלנו ? כמה פנוי ?

**משימה 2**

* + - 1. תסתכלו על תוכנית ותסבירו במספר משפטים מה היא עושה **memory\_use.c**
      2. תמלאו את **MEM / USED** (שמציגה פקודת FREE) בטבלה הבאה, עבור 3 הרצות השונות ושלבים שונים של התוכנית:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| אחרי שחרור | אחרי מילויי | אחרי הקצאה | לפני הרצה | גודל זיכרון (ב-B) | הרצה |
|  |  |  |  | 1000 | 1 |
|  |  |  |  | 5000000 | 2 |
|  |  |  |  | 10000000 | 3 |

* + - 1. תריצו תוכנית שוב עד לפני שלב שחרור הזיכרון )תקצו זיכרון בגודל ( INT 1000.000.000

תציגו צריכת זיכרון ו-CPU כל כמה שניות עיי' פקודה הבאה:

**ps -p <PID> -o %cpu**  (ה-PID הוא של הרצת התוכנית)

מה אתם רואים? תנסו להסביר את התוצאות**. רמז:** ה-CPU מציג את האחוז היחסי של זמן ריצה