

OS (Linux)

תוכנית לימודים במקצוע הנדסת תוכנה לחטיבה העליונה

מערכות הפעלה מודרניות בסביבת Linux Ver. 2.0

התכנית פותחה על ידי מר יוסף בלאן, חבר ועדת המקצוע. אושרה על ידי ועדת המקצוע.

OS (Linux)

press on the table

פרקי הלימוד

שעות לימוד	שם הפרק	מספר הפרק
32	היכרות עם מערכת ההפעלה Linux.	1
40	תכנות מתקדם בשפת java.	2
12	מודל OSI.	3
70	תכנות בסביבת Java SWING עבור Linux.	4
12	תכנות בעזרת מנגנון האירועים Event-Driven Programming	5
70	אלגוריתמים ותכנות מקבילי במערכות ההפעלה Linux	6
20	תכנות מקבילי וסנכרון ממשק המשתמש בלינוקס Multithreading GUI	7
16	שרת בסיס נתונים MySQL ו-JDBC	8
30	ממשקי תקשורת שרת/לקוח TCP/IP ובניית Multithreaded TCP/IP Client/Server.	9
36	תקשורת TCP/IP עם Android כמערכת הפעלה אחרת בסביבת Java	10
16	טיפול בקבצים Files and Stream.	11
16	העברת מידע בצורת XML ו-JSON.	12
80	פרויקט	
450	סה"כ	

OS (Linux)

שכבה	נושא	#
יא	תכנות מתקדם : (OOP, Try..catch, Exceptions)	1
יא	GUI (Swing), Threading	2
יא	Files, Streams	3
יב	Networking, OSI	4
יב	מבנה מחשב ומערכת הפעלה	5
יב	Linux: bash, file system, ,scripts	6
יב	TCP/IP, UDP/IP Socket Networking in Python. Multithreaded server/client	7
יב	טכנולוגיה MySQLdb לעבודה עם בסיס הנתונים	8
יב	תקשורת עם סמארטפון אנדרואיד	9

- מעקב אחרי ילדים של אלמז רועי
- סל קניות משותף של דור גרוס
- מעקב רפואי של אביאתר פרקר
- משחקי רשת
- שילוב מוסיקה עם צבעים עבור דיסקוטק
- מבחן ממוחשב
- מערכת למידה

OS (Linux)

שכבה יא

מימוש תוכנית בזיכרון הפנימי

- כל תוכנית בריצה מייצרת איזור בזיכרון הפנימי של המחשב (RAM).
 - האיזור הזה נקרה "תהליך" או באנגלית Process.
 - כל תהליך מחולק ל-4 תתי האיזורים - segments:
- code, data, stack, heap

code

מכיל את התוכנית עצמה: מחלקות ותוכנית ראשית. כל תוכנית עוברת תרגום לשפת CPU לפני שהיא נכנסת לסגמנט code.

data •

מכיל משתנים סטטיים

stack •

אחראי על ביצוע פעולות

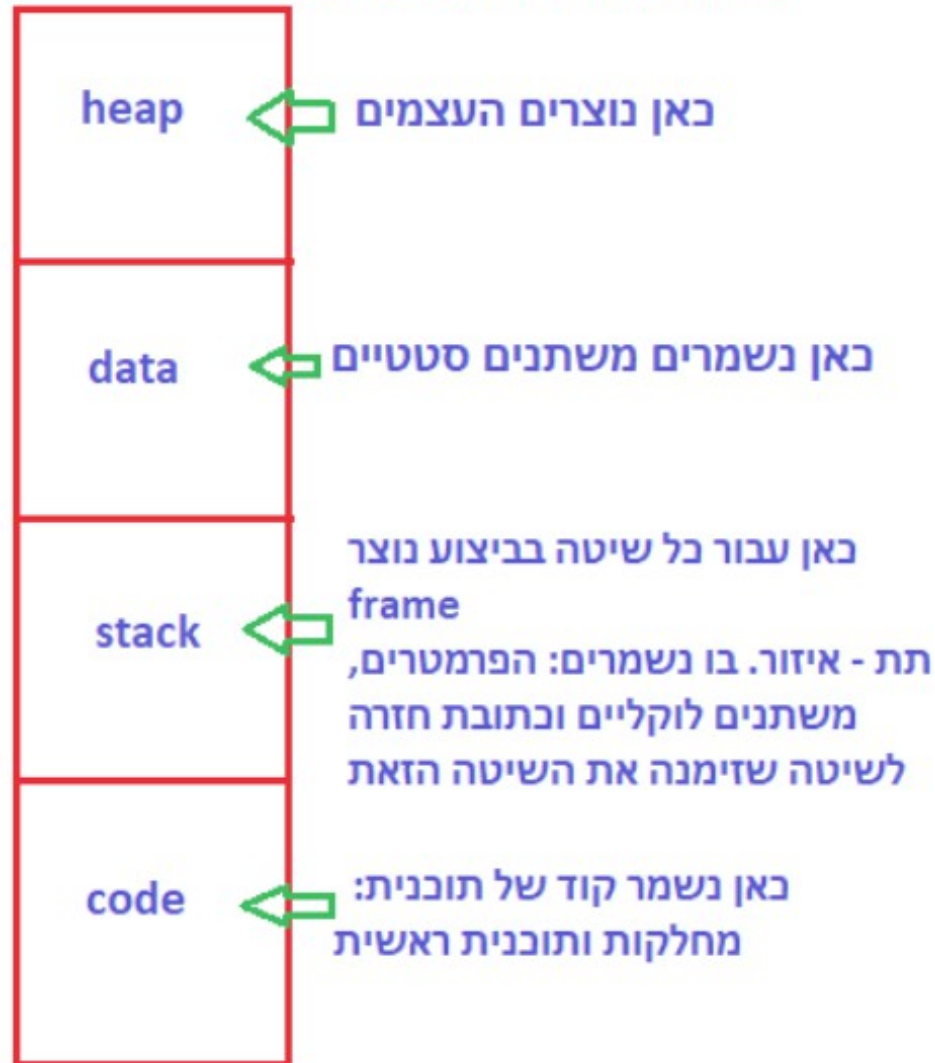
heap •

הינו איזור שבו נוצרים העצמים

process

תהליך

מבנה התהליך: תוכנית בריצה.



עצם ואופרטור new

- רק האופרטור **new** מייצר עצמים. הוא מייצר אותם באיזור heap. הוא מקצה מספר תאי הזיכרון שמספיקים לשמירת כל התכונות שהוגדרו במחלקה + כתובות כל השיטות סטטיות ולא סטטיות ומשתנים סטטיים. לכן ניתן לגשת לכל איברי המחלקה דרך העצם.
- כתובות השיטות והמשתנים הסטטיים הינם שווים עבור כל העצמים ממחלקה אחת. לכן נצייר את העצמים רק כאוסף תכונות.
- נתבונן בקוד הבא שבתוכנית הראשית:

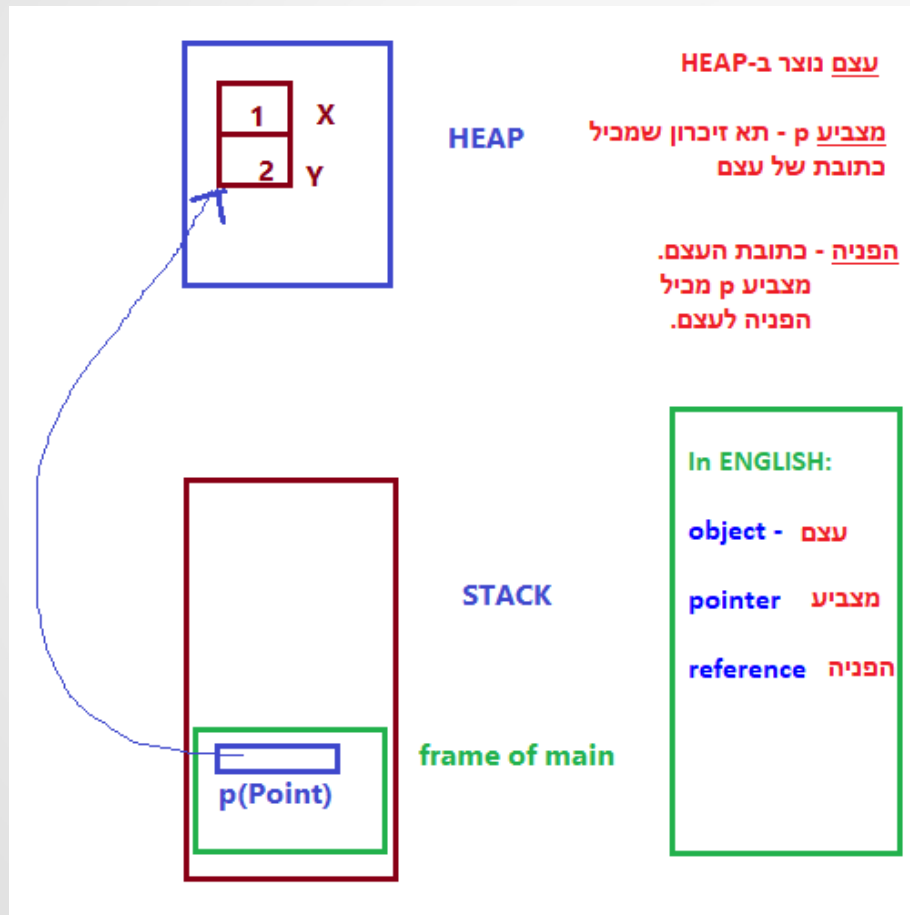
```
Point p = new Point();
```

```
p.setX(1)
```

```
p.setY(2)
```

ציור העצם שנוצר מופיע בשקופית הבאה. המצביע p נוצר באיזור stack בתוך frame של התוכנית הראשית.

עצם ואופרטור new

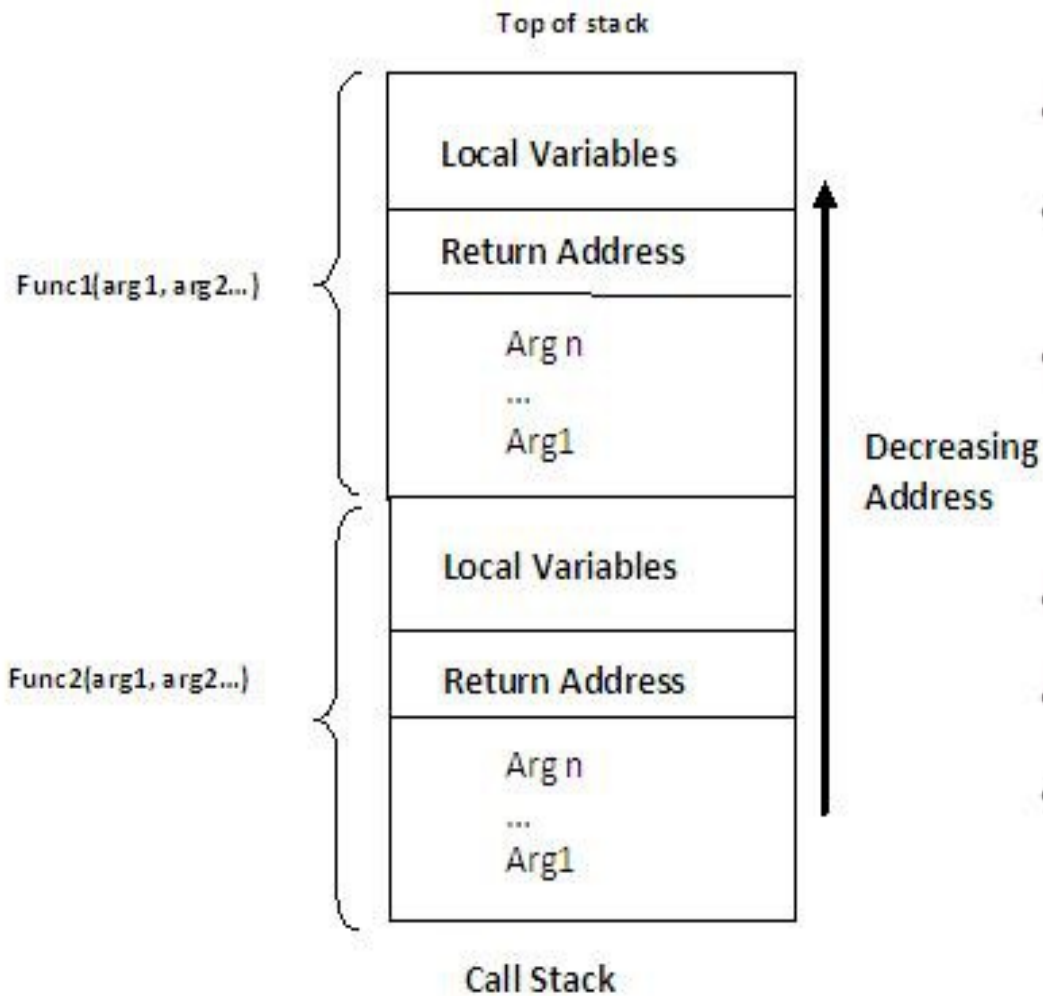


- ה-frame של התוכנית הראשית ימחק אחרי סיום ביצועה. באותו הרגע יסתיימו החיים של המצביע p. ואז העצם ישאר ללא קישור אליו.
- קיים מנגנון למחיקת עצמים שנשארו ללא קשר אליהם.

שמו garbage collection

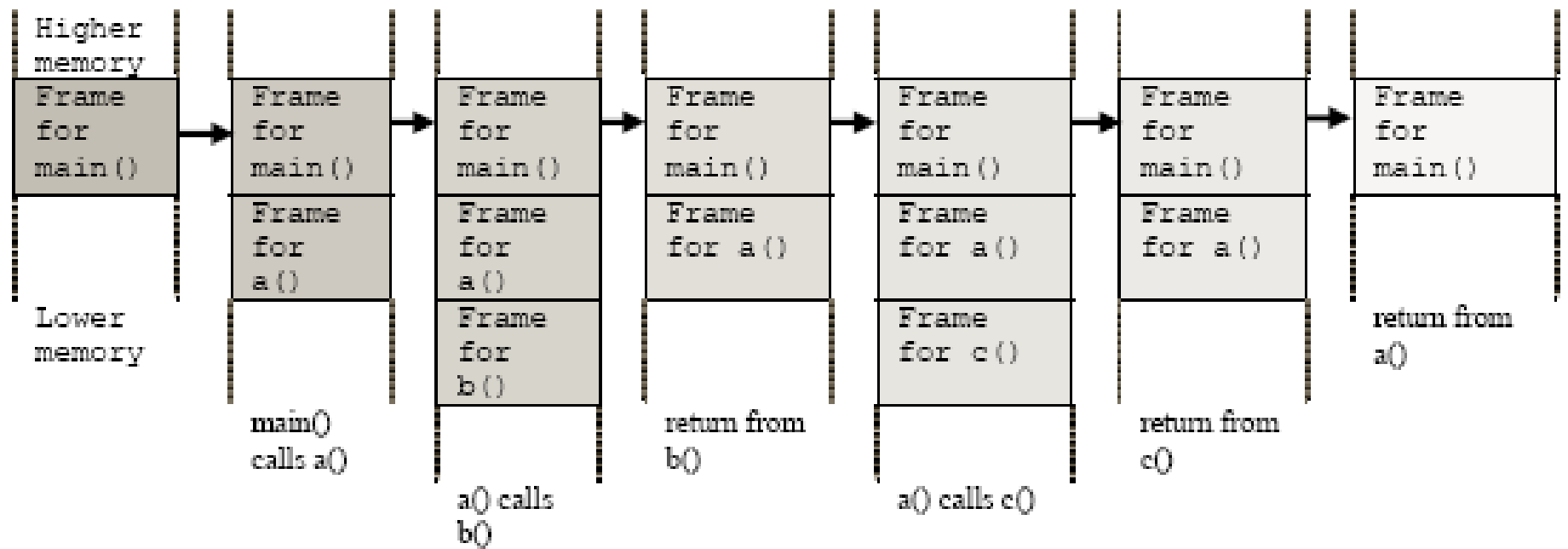
- הפעולות של המנגנון הזה רצות במקביל לביצוע התוכניות. הן מחפשות עצמים ב-heap שנשארו ללא מצביעים ומשמידים אותם: מחזירים את הזיכרון של העצם למפת הזיכרון הפנוי.

פעולות מתבצעות בעזרת-stack

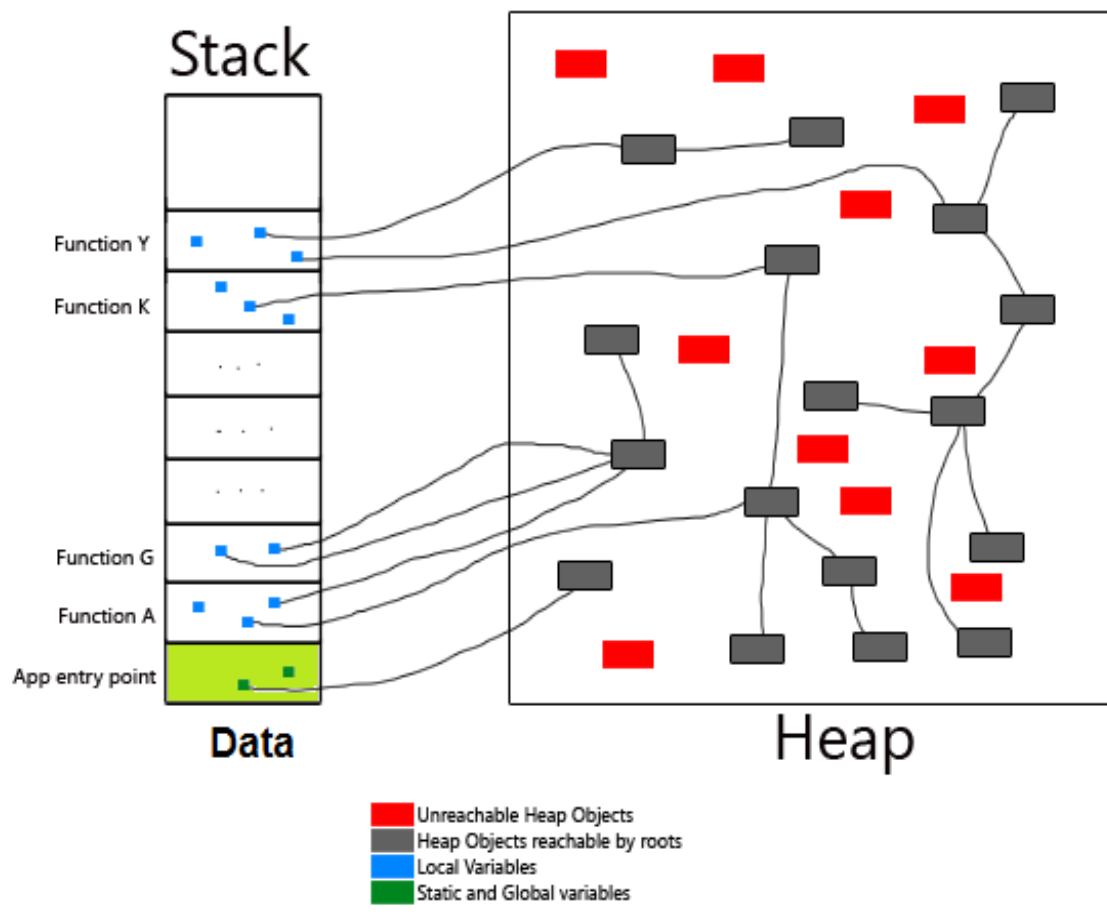


- **Call Stack:** is a stack that contains a collection of stack frames.
- A **stack frame**, also called as **activation record**, contains information about function call.
- When the program runs, the call stack contains at least one stack frame which is for the main function. A stack frame is created whenever a function is called.
- Arguments are inserted into stack frame from *right to left*.
- *The stack grows from higher addresses to lower addresses.*
- A **stack pointer register** (register 1) is used to mark the current "top" of the stack.

פעולות מתבצעות בעזרת-stack



העצמים נוצרים ב-heap



- המצביעים שמכילים את כתובות העצמים הם נוצרים ב:
 1. stack.
 2. data.
 3. heap.אם מדובר על משתנה סטטי עצם יכול להכיל תכונה שהיא כתובת של עצם אחר. במקרה הזה המצביע נמצא באיזור העצם: ב-heap.

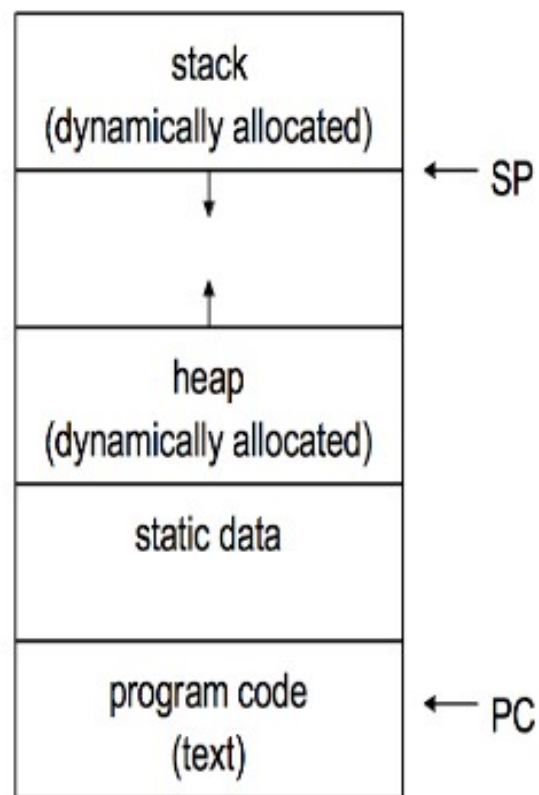
ריבוי תהליכים

Process Control Block (PCB)

pointer	process state
process number	
program counter	
registers	
memory limits	
open files	
Etc...	

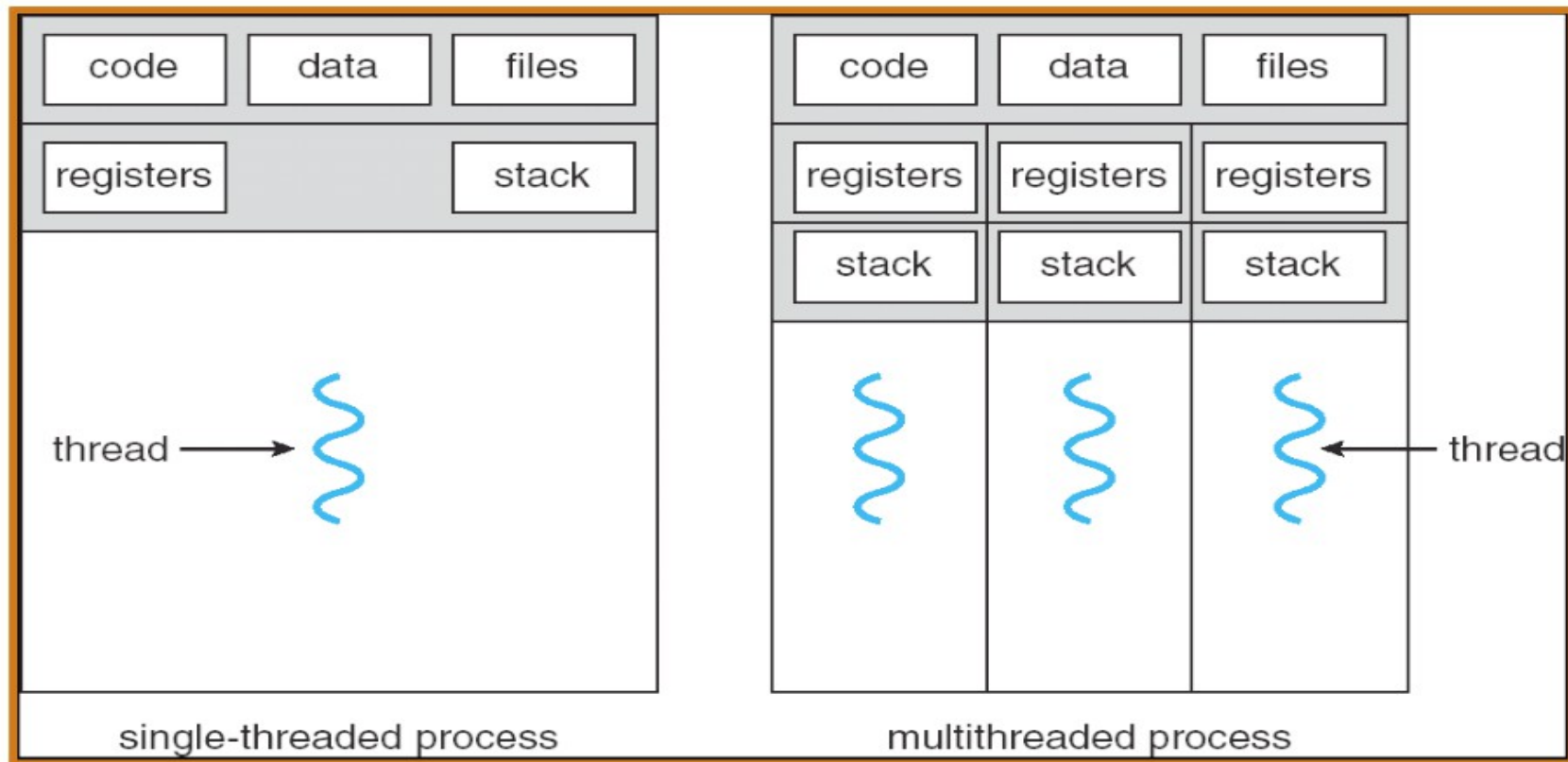
Address Space Linux-Windows process

0xFFFFFFFF
↑
Virtual
address space
↓
0x00000000



תהליך ותהליכון

Single and Multithreaded Processes



Role Thread in GUI Programming

JVM runs some parallel threads automatic

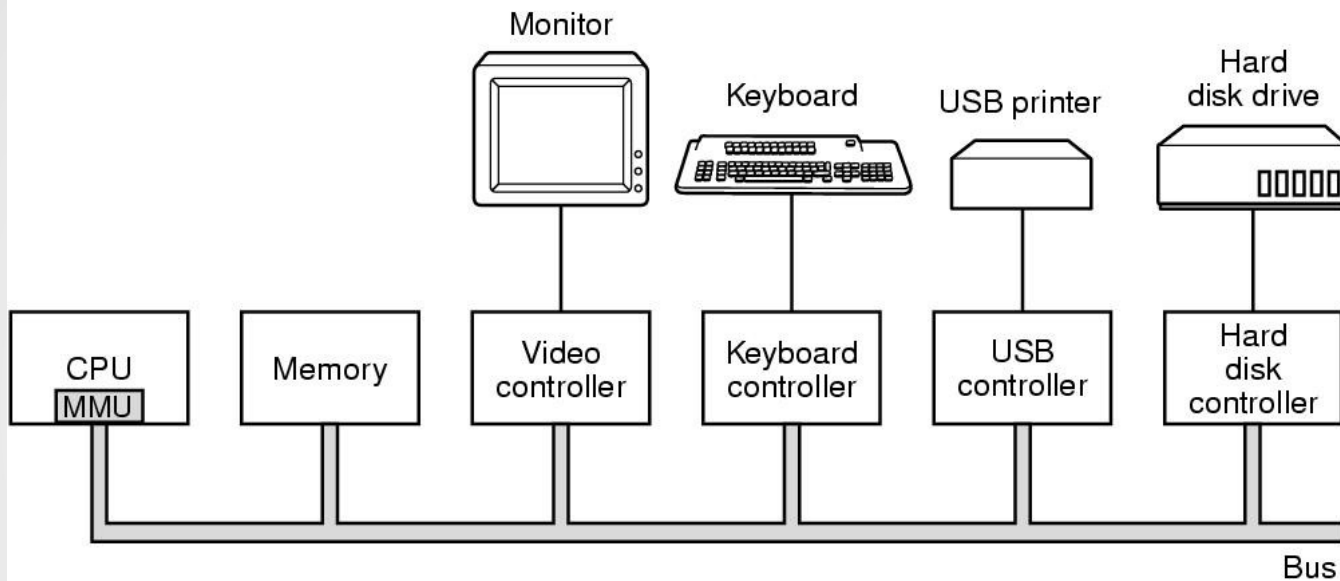
מבלי לשאול אותנו

- הרצה של תוכנית ראשית - **main thread**
- טיפול בציור מסך ותגובות לאירועים - **awt-event thread**
- **איסוף זבל** - באנגלית: Garbage collection - הוא תהליך שבו סביבת הריצה של תוכנית מחשב מנקה מהזיכרון את האובייקטים שאין בהם עוד צורך.
- עבודה עם ערוץ קלט/פלט כמו טעינה של תמונה, כתיבה לתוך קובץ, קריאה/כתיבה ל-socket

OS (Linux)

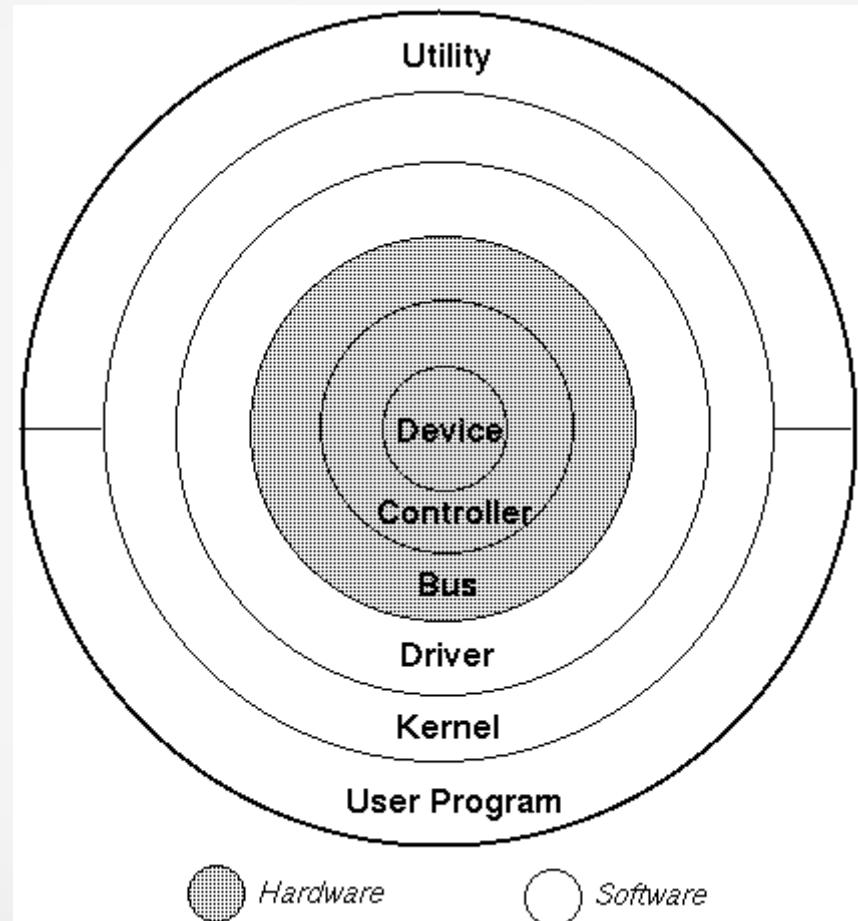
שכבה יב

Computer Hardware Review

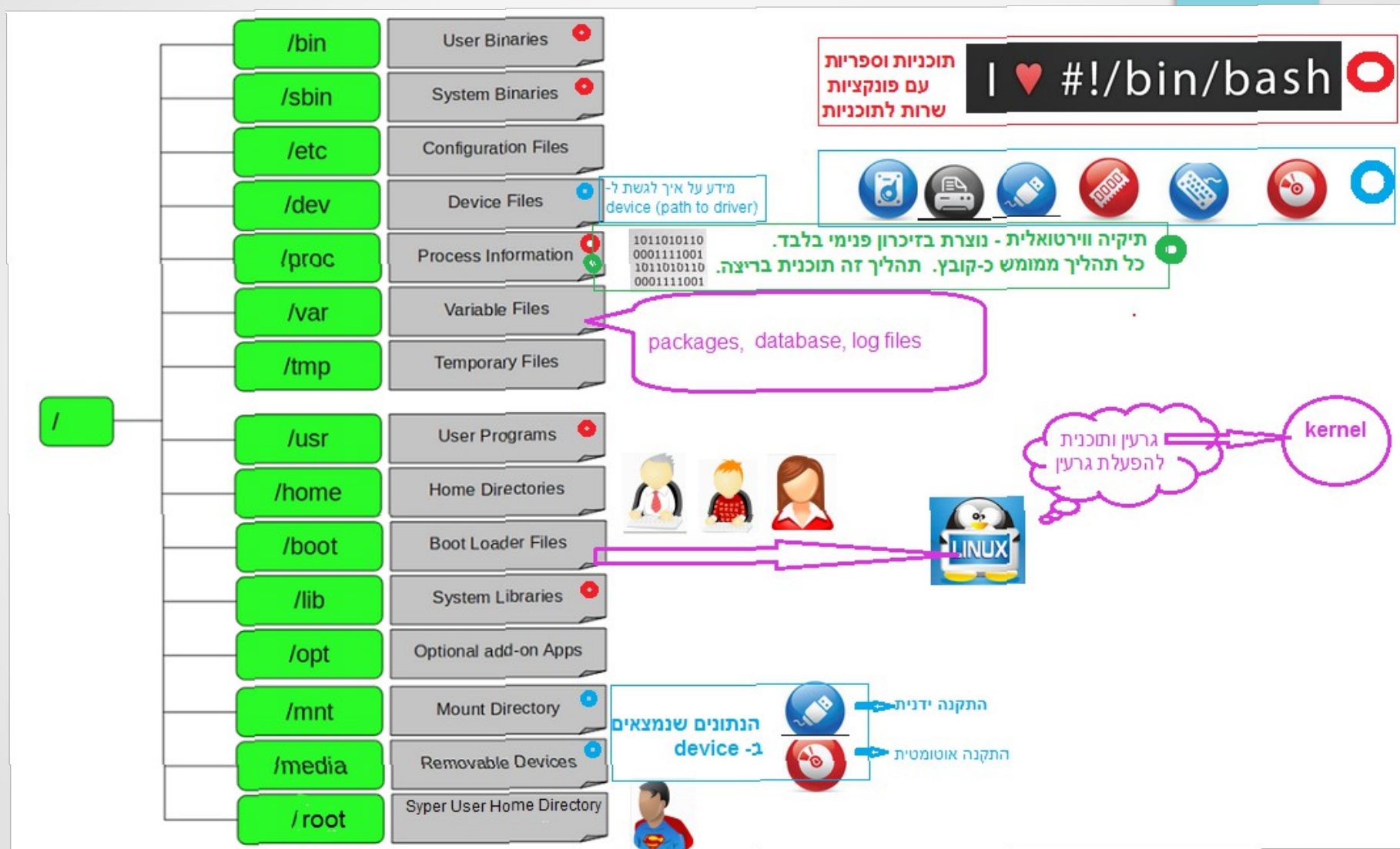


Some of the components of a simple personal computer.

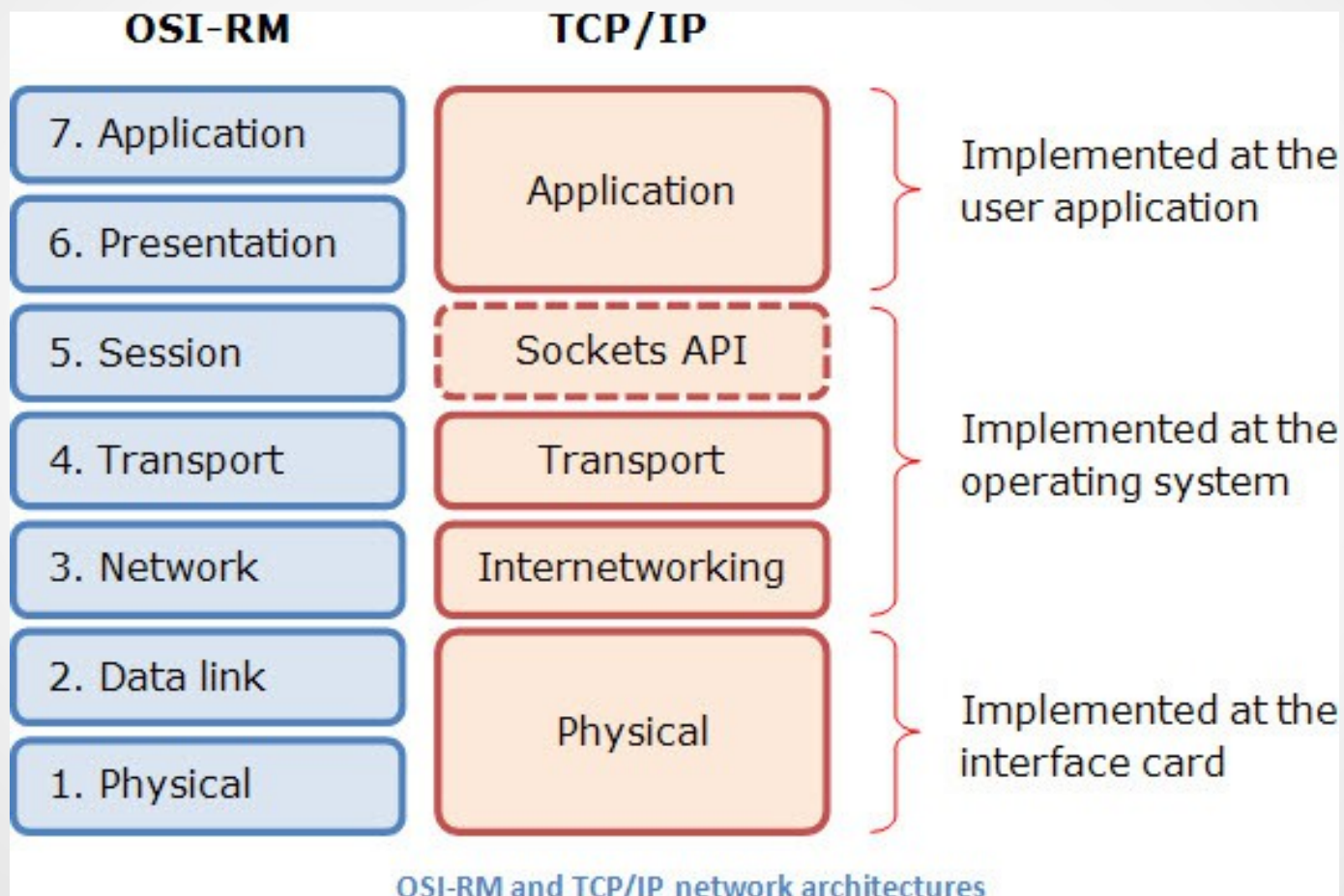
Software and Hardware Layers



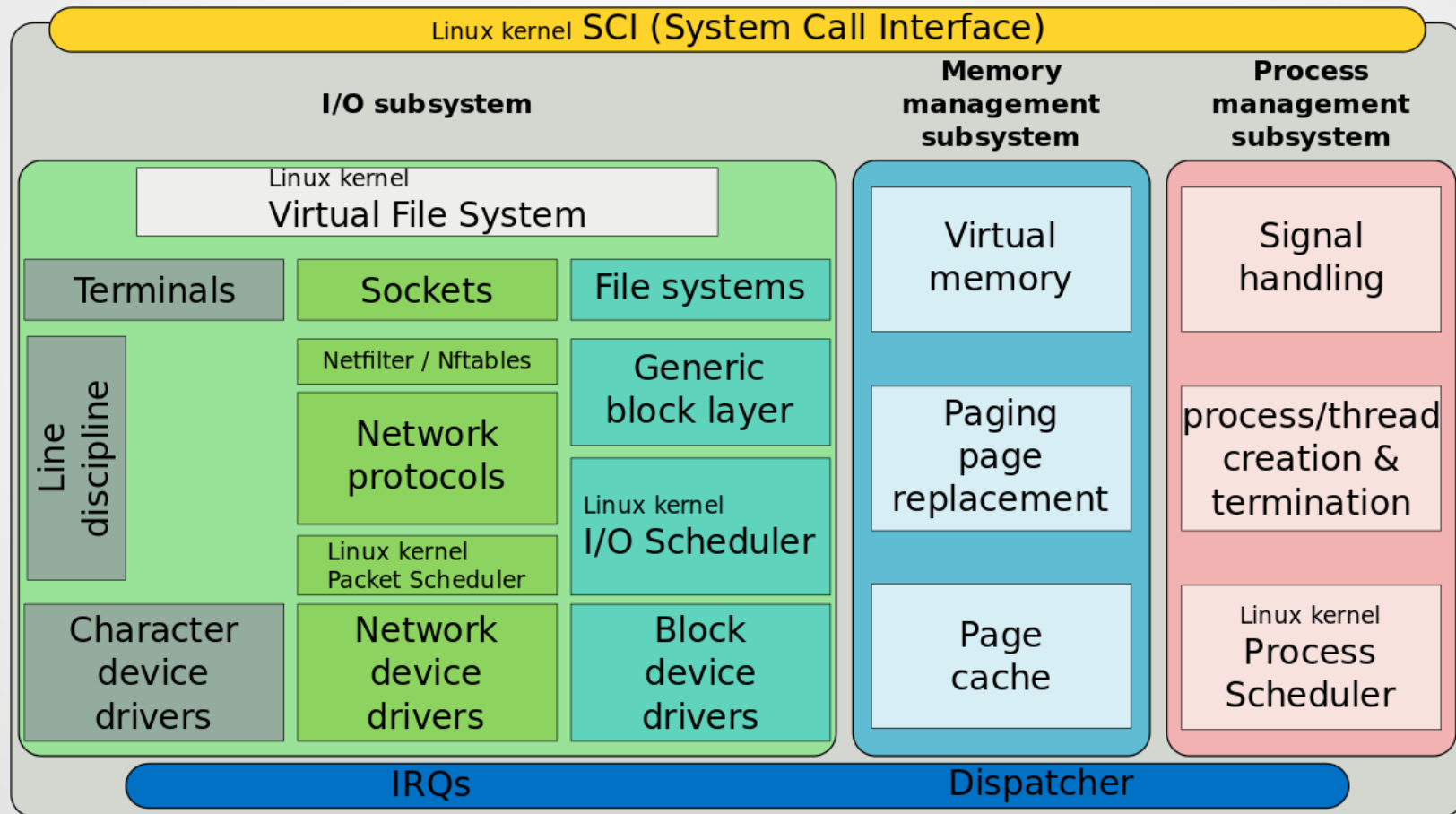
Filesystem Hierarchy Standard (FHS)



איך רשת עובדת

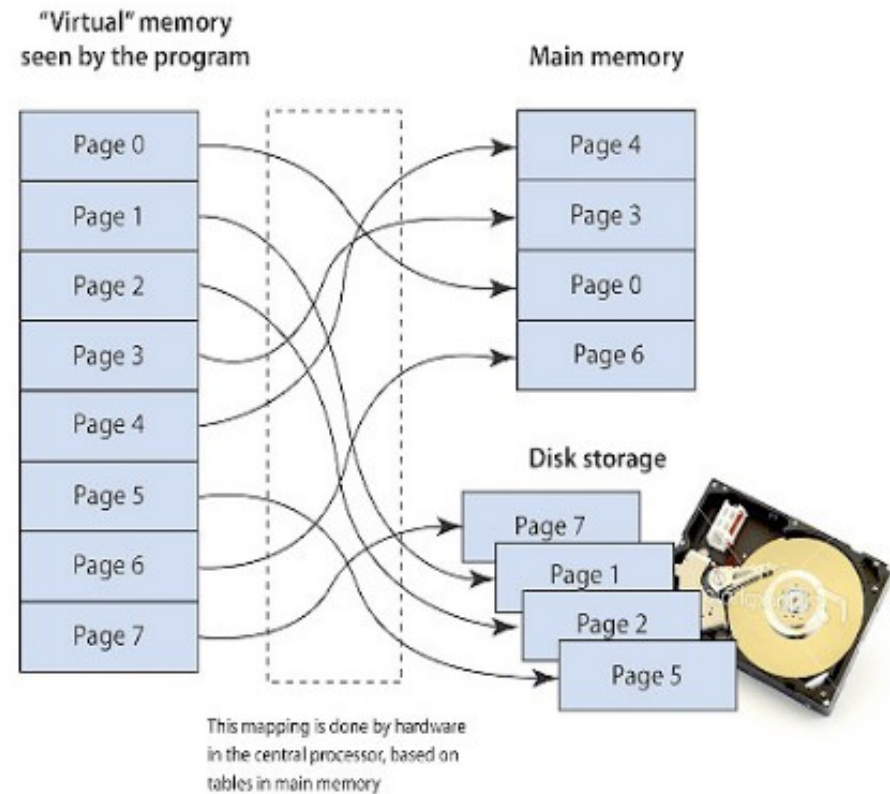


תפקיד של גרעין

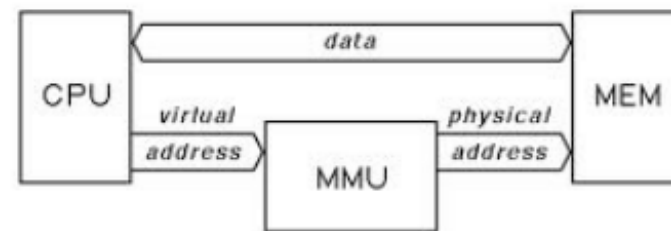
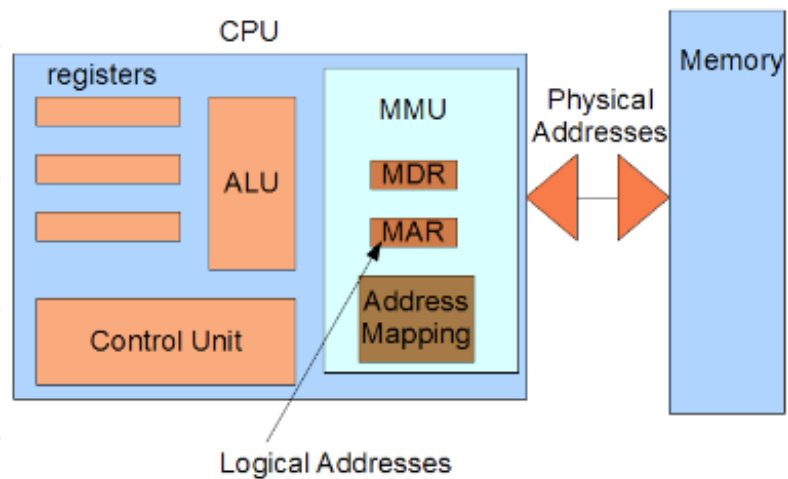


Virtual memory

- המעבד "חושב" שבזיכרון הפנימי יש כל 7 דפים. אבל חלק מהם נמצאים בזיכרון החיצוני. הם נטענים לזיכרון הפנימי רק במקרה והמעבד ניגש לאחת מהכתובות בהם.



Virtual memory



MMU – Memory Management Unit

זה רכיב שתפקידו להמיר כתובת לוגית (virtual) של תוכנית או נתונים לכתובת פיזית שנמצאת בזיכרון הפנימי.