Overview

לרוב, תעשיית המחשבים שמה הדגש על פיתוח מהר של מוצרי תוכנה, שימוש חוזר של קוד ודפוסי תכנות, ותאימות עם מערכות תוכנה קיימות. בניגוד למגמה הזאת, בתחומים מסוימים במקצוע שמים דגש על הפקת קוד יעיל ומהיר – תכנות מתמטית, תכנות לתשתית תקשורת, ובמיוחד תעשיית המשחקים.

הקורס **ארכיטקטורות מחשבים** עוסק בשיקולים וחשיבה מאחורי התכנון של מעבדים למחשבים, טאבלטים, וסמארטפונים, ומלמד שיטות לשיפור (קיצור) זמן ריצה של תוכניות שלמות או קטעי קוד קריטיים כמודול במערכת גדולה יותר. ההרצאות משלבות תיאוריה עם דוגמאות מעשיות במטרה להבין את הפלטפורמות העדכניות. לעבודה מסכמת תכתבו תוכניות C, תקמפלו לשפת סף, ותתקנו את קוד הסף על מנת לשפר את הביצועים.

הקורס מחולק לעשרה פרקים:

1. **מבוא לחקר ביצועים**: נציג מושגי יסוד בביצועי מחשבים וגורמי השפעה בחומרה ותוכנה
2. **רענון ארגון המחשב**: נציג מבנה von Neumann והמרכיבים שלו — CPU, ALU, זיכרון, פקודות, קלט/פלט ובקרה. גם נדבר על תכנון קבוצת פקודות :סוגי פקודה ,מבני פקודות , שיטות מיעון, והגישה הקלאסית לתכנון מחשבים מורכבים.
3. **תורת הביצועים המתמטית**: מדידה והשוואה של ביצועים, benchmarks, חוק Amdahl, משוואת ה- CPU וכלים שונים לקביעת הביצועים של מעבד.
4. **תכנון קבוצת פקודות מסוג RISC**. בפרק הזה נלמד את השיקולים לתכנון מעבדים מודרניים ושיפור הביצועים ביחס ממחשבים הקלאסיים מסוג CISC.
5. **מודל ה- DLX**: נלמד גרסה בסיסית למעבד מסוג RISC ושיטות המימוש שלו.
6. **שיפור מודל ה- DLX**: נציג כיצד ארכיטקטים מנתחים ביצועי מעבד בזמן ריצה ומכניסים שיפורים כמן חפיפה (pipelining) לתכנון קיים.
7. **חשבון מחשב**: הצגת נתונים ,חיבור וחיסור ,כפל ,חילוק ,חשבון במספרים צפים
8. **היראַרכיה של זיכרון**: אוגרים, מטמון (cache), אחסון לטווח ארוך. נציג שיטות ניהול זכרון והשיקולים בתכנון מערכות זכרון קיימים. גם נלמד על מערכות קלט/פלט.
9. **מערכות מודרניות**: superscalar, dynamic scheduling, superpipeline, multicore, parallel programming ועוד שיטות עדכניות במעבדים של היום.
10. **RISC בשוק**: סקירה על מעבדי RISC מסחריים כמו MIPS (הבסיס למודל DLX) ומעבד ARM שקיים ביותר מ- 95% מהטלפונים והטבלטים.

מי שמסיים את הקורס בהצלחה יהיה מסוגל להסביר את הפיתוח של מחשבים והארכיטקטורה שלהם משנות ה- 1970 עד לטכנולוגיות האחרונות. הקורס יקל על לימוד טכנולוגיות מתפתחות חדשות, וקבלת החלטות נבונות בבחירת מוצרים. הקורס גם מספק שפה משותפת לאנשי תוכנה ומהנדסי חומרה שתקל על שיתוף פעולה במחקר ופתוח.

הקורס בנוי על שילוב של מצגות וסיכומים כתובים בטקסט מלא. מטרת המצגות היא להציג איורים, תוכניות, והגדרה בסיסית של מושגים בקיצור נמרץ. השקפים לא מהווים תחליף לקריאה בסיכומים וביצוע תרגילים. <u>לימוד מהשקפים לבדם לא מומלץ</u>.

<u>המלצות ללימוד עצמי</u>

כפי שניתן לראות באתר הקורס, מבנה הלימוד מוגדר בפריסה שבועית של הנושאים והתרגילים. קורס מתוקשב מתבסס על הציפייה שסטודנטים בשלב מתקדם מכירים את שיטות הלימוד המוצלחות להם ויודעים להפעיל את המשמעת העצמית הדרושה ללמוד באופן שוטף ולהגיש תרגילים בהתאם למועדי ההגשה. אבל מהניסיון, הפיתוי לדחות לימוד קורס מתוקשב חזק (עד כדי להשאיר אותו לתקופת המבחנים). הכניעה לפיתוי זה לא מביאה לתוצאות מספקות. ממבט ראשון, נדמה שיש יותר מדי נושאים בקורס ללמוד בסמסטר אחד. אבל כשלומדים באופן עקבי וסופגים לאט, רואים שבעצם מדובר במספר לא גדול של מושגים קשורים. כל שיטה מהווה שיפור מדורג על שיטה הקודמת ברצף והנושאים מתגלגלים אחד לשני באופן טבעי.

בכל נושא, מומלץ לעבור על המצגות עם הסיכומים (הרצאות מודפסות) פתוחים ועין על התרגיל שמיועד להשלים את נושא על ידי תרגול. חלק מהנושאים קל לקלוט מהמצגת, ובחלק אחר חשוב לקרוא את הטקסט הנלווה כדי להבין את ההקשר.

בכל מקרה שמשהו עדיין לא ברור תפנה/י אלי להסבר נוסף בדואר אלקטרוני (martin@hac.ac.il) ואשתדל לענות לכל הודעה תוך 24 שעות.