

מודל ה RAM

המודל דומה למודל של מחשב רגיל פרט לשני הבדלים:

1. כמות הזיכרון שלו היא אינסופית.
2. כל משתנה יכול להכיל כל מספר בלי הגבלה של גודל הייצוג.

התוכנית מתחילה מכתובת אפס עד כתובת חיובית כלשהי, ובנוסף יש עוד כתובת חיוביות ושליליות. יש n רגיסטרים - $r_1, r_2, r_3, \dots, r_n$ כאשר לשלושת הראשונים יש תפקידים מיוחדים:

$$r_1 = pc$$

$$r_2 = input$$

$$r_3 = output$$

בתחילת התוכנית יש אפסים בכל הזיכרון ובכל הרגיסטרים פרט למקום בזיכרון בו כתובה התוכנית, ופרט לרגיסטר r_2 בו רשום הקלט.

צעד החישוב: כמו בכל מחשב רגיל: קוראים את הפקודה מהזיכרון בכתובת שרשומה ב pc , מגדילים את ה pc באחד, מבצעים את הפקודה, וחוזרים על כך עד שמגיעים לפקודה $halt$.

הערות:

1. התוכנית מוגדרת כחלק מהמכונה (כמו פונקציית מעברים במכונת טיורינג). תוכניות שונות מגדירות מכונות שונות.

2. כל מחרוזת סופית מעל כל א"ב ניתנת לייצוג בבסיס בינארי כמספר שלם: $|Z| = |\Sigma^*| = \aleph_0$.

3. מאפשרים מספרים שליליים וכתובות שליליות כך שהפקודה dec לא תיכשל.

4. תוכן כל תא אינו חסום כדי לאפשר גישה לזיכרון (האינסופי)

5. הדרישה למספר סופי של רגיסטרים נועדה להקל. ניתן לסמלץ אינסוף רגיסטרים באמצעות הזיכרון.

6. סט הפקודות הנ"ל מייצג אופציה אחת לסט פקודות. אפשר להחליפו לסט פקודות אחר.

טענה: מודל ה RAM שקול למודל מכונת טיורינג.

כיוון קל: דף העזר מראה כיצד לסמלץ מכונת טיורינג בשפה דמוית C. ניתן "לקמפל" תוכנית זו "לאסמבלי" של מודל ה RAM בפעולת קומפילציה פשוטה.

כיוון קשה:

נשתמש במ"ט עם $n+3$ סרטים.

n סרטים עבור הרגיסטרים, עוד סרט אחד בשם "זיכרון", עוד סרט אחד בשם "כתובת", עוד סרט אחד בשם "תוכן".

נתחיל בהעתקת הקלט מהסרט הראשון (r_1) לסרט השני (r_2), ונרשום אפס בכל שאר הרגיסטרים

($r_1, r_3 - r_n$) במקום הבלנקים ובמקום הקלט.

בתחילת הסרט של הזיכרון נסמן @, ואחריו, עבור כל כתובת נרשום את הכתובת, אח"כ \$ ואח"כ

התוכן ואח"כ #

כאשר רוצים לקרוא תוכן של כתובת כלשהי בזיכרון, רושמים את הכתובת בסרט הכתובת, ומתחילים להשוות לסרט הזיכרון, עד שמוציאים את הכתובת הזוהה בסרט הזיכרון, ואז מעתיקים את התוכן ממנו אל סרט התוכן.

אם מגיעים לבלנקים, סימן שרוצים לקרוא מכתובת שעדיין לא נכתב אליה אף פעם ולכן יש בה אפס.

כאשר רוצים לכתוב תוכן לכתובת כלשהי בזיכרון, רושמים את הכתובת בסרט הכתובת, ומתחילים

להשוות לסרט הזיכרון. אם מוציאים בדרך את הכתובת, רושמים במקומה ובמקום התוכן X-ים,

וכשמגיעים לסוף הסרט (הבלנקים) רושמים שם את הכתובת ואת התוכן עם התווים \$ ו #.

באתחול, נעתיק את תוכן התוכנית אל הזיכרון.
 אח"כ מבצעים צעדי חישוב.
 אם מגיעים לפקודת $halt$, מעתיקים מ r_3 ל r_1 את הפלט.

כיצד מבצעים צעדי חישוב?

1. מעתיקים pc לסרט הכתובת ומבצעים פעולת קריאה, כפי שתואר קודם.
2. מוסיפים ל pc אחד, כפי שלמדנו לבצע בתרגול השני.
3. "נפענח" את הפקודה.

לדוגמה פקודת: $load \quad r_i \leftarrow (r_j) +$

4. נבצע את הפקודה: נעתיק את r_j לסרט הכתובת, ונבצע פעולת קריאה.
5. נעתיק את סרט התוכן ל r_i .
6. נקדם את r_j באחד, כפי שלמדנו לבצע בתרגול השני.

לדוגמה פקודת: $store \quad r_i \rightarrow (r_j)$

4. נבצע את הפקודה: נעתיק את r_i לסרט התוכן.
5. נעתיק את r_j לסרט הכתובת.
6. נבצע פקודת כתיבה.