בס"ד

שם : רון אמסלם
תז:326029600

תכנות מערכות 1 – מטלה 1

חלק ראשון

1. קובץ בעל סיומת h מכיל בתוכו את כל ההצהרות של הפונקציות שנעשה בהם שימוש בעת הרצת תוכנית

קובץ בעל סיומת c מכיל את המימוש של הפונקציות שהוכרזו בקובץ h.

הם עוזרים לנו לשמור על עקרונות תכנות נכונים בכך ש

- קובץ H מספק "ממשק" שמאפשר למשתמשים אחרים להשתמש בהצהרות שונות ללא להכיר את המימוש של הפונקציה.
- קובץ C שומר את הפרטים הפנימיים מוסתרים.
- שינוי במימוש בקובץ C לא מחייב שינוי בקבצי H שממשתמשים בו.
- ניתן לעדכן פונקציות בקלות מבלי להפריע לקבצים אחרים.

2. בשלב הראשון הpreprocessor עובר על קובץ הטקסט ועושה עליו פעולות טקסטואליות, למשל הוא לוקח את הקובץ h. ומעתיק אותו לקובץ המקורי, אותו דבר לגבי studio, הוא מוריד הערות, וכל זה תוך כדי שהוא עובד ברמת הטקסט.

בשלב השני הקומפיילר לוקח את הקוד הנמצא בקובץ הטסט עם סיומת c והופך אותו לשפת מכונה, דבר זה קורה עבור כל קובץ בנפרד, כמו כן הקומפיילר רץ קדימה לא חוזר אחורה . בנוסף נוצר קובץ בשפה בינארית עם סיומת o

בשלב השלישי יש לנו את הlinker שהוא מחבר את כל קבצי ה-O לקובץ אחד גדול ומפנה כל דבר לאן שצריך בנוסף הלינקר מוודא שלכל הפונקציות יש מימוש והצהרה בכך שהוא יודע לקשר בין הצהרה לבין מימוש . בסיום התהליך אנו נקבל קובץ out. .

3. בשלב הlinker יצירת קובץ ההרצה נכנסות הספריות. המקשר (Linker) לוקח את כל קובצי האובייקט שנוצרו בהידור ומחבר אותם יחד עם הפונקציות הרלוונטיות מהספריות הדרושות.

4. יתרון של ספרייה דינמית על פני ספרייה סטטית:

מבחינת חיסכון בזיכרון ספריות דינמיות נטענות לזיכרון רק בזמן ריצה, ולכן קובץ ההרצה עצמו קטן יותר. כלומר ספריות סטטיות גדולות בהרבה , מכיוון שהתוכניות חיצוניות בניות בקובץ ההפעלה לעומת ספרייה דינמית שהיא קטנה בהרבה מכיוון שיש רק עותק אחד של ספרייה דינמית הנשמר בזיכרון .

יתרון של ספרייה סטטית על פני ספרייה דינמית:
ספריות סטטיות מוטמעות בקובץ ההרצה, מה שהופך אותו לתוכנית עצמאית שאינה תלויה בקובצי ספרייה חיצוניים בזמן הריצה. כלומר זמן הקישור של ספריה סטטית קורה בשלב האחרון של תהליך הקומפילציה לאחר שתוכנית ממוקמת בזיכרון לעומת ספריה דינמית שמתווספות במהלך תהליך הקישור כאשר קצבי ההפעלה וספריות מתווספים לזיכרון.

מתי נרצה להשתמש בכל אחת?

בספריות דינמיות לרוב נרצה להשתמש כאשר יש בתוכניות מורכבות שמשתמשות בהרבה ספריות גדולות, כדי לחסוך מקום. וספריות סטטיות נרצה להשתמש כאשר התוכניות קטנות או תוכניות שמיועדות להפצה למשתמשים רבים, ללא תלות בסביבת היעד.

5. קבצי (Object Files) OBJ.שנוצרים בשפת C במהלך תהליך ההידור כתובים בפורמט בינארי.

6. הסיומות עבור קבצי ספריות דינמיות הוא סיומת so וסטטיות בסיומת a.

בס"ד

חלק שני:

1.נשתמש בדגל Wall-

2. כן ניתן לסטות מהשם הקבוע של קובץ MAKEFILE ולתת שם לבחירתכם פשוט נרשום בטרמינל
make -f nameMakeFile
3. כדי להשתמש במשתנה, יש לעטוף את שמו בסוגריים עם $, לדוגמא כך $(VARIABLE_NAME).
4. החוקים התקיימו אם
היעד לא קיים , היעד ישן יותר מהתלות , התלות של היעד דורשת עדכון ,חוק מופעל ידנית על ידי המשתמש.

5. משום ש Makefile פועל על בסיס תלות ועדכוני זמן, ומבצע רק את הפעולות הדרושות לעדכון היעדים, מה שהופך אותו ליעיל מאוד בניהול תהליכי בנייה מורכבים.

6. נוצרים חוקים עקיפים כאשר אין חוק מפורש מוגדר עבור Target
אם לא סיפקת חוק שמסביר כיצד לבנות את היעד make ,ינסה להפעיל חוק מובנה שמתאים.
הבעיה בהם שקשה להבין מה בדיוק עושה make, במיוחד בפרויקטים מורכבים.

7. שימוש בדגל MM- , לדוגמא gcc -MM main.c
8.צריך להשתמש בפקודה ar, שהיא הכלי המשמש ליצירת, עדכון וניהול של ספריות סטטיות.
ar rcs libname.a file1.o file2.o file3.o

חלק 3:

השלב בו אנחנו מדלגים בין הקימפול ללינקוג' הוא שלב האסמבלר ,שבו קוד האסמבלי מתורגם לקוד אובייקט שניתן לבצע עליו לינקוג.'