

30 אוגוסט 2023

תרגיל תכנות שלישי במחשוב מקבילי ומבוזר (cuda) סמסטר קיץ 2023

התוכנית תשתמש ב-cuda ותרוץ על Linux. הגשה (בזוגות או ביחידים) דרך moodle.

יש לכתוב פונקציית kernel המשווה בין 2 מחרוזות ומחזירה את תוצאת ההשוואה כמספר שלם בדומה לפונקציית strcmp :

אם המחרוזות שוות היא מחזירה אפס. אם המחרוזות הראשונה גדולה יותר היא תחזיר מספר חיובי. אם המחרוזות הראשונה קטנה יותר היא תחזיר מספר שלילי.

יש להיעזר בפעולת scan (prefix sum) כדי לממש את השוואת המחרוזות.

לתרגיל מצורף קובץ בשם scan_strcmp_starter.cu הכולל את כל התכנית מלבד הקוד של ה-kernel הנקרא my_strcmp אותו עליכם להשלים. זה אומר שצריך לכתוב בסביבות 15 שורות קוד נטו. יש להגיש את הקובץ עם התוספות שלכם. לקובץ זה קראו פשוט scan_strcmp.cu (ללא התוספת "_starter").

הפונקציית my_strcmp (kernel)

הנה החתימה שלה :

```
void my_strcmp(const char *s1, int n1, const char *s2, int n2, int *result)
```

s1 ו-s2 הן המחרוזות שיש להשוות. בסוף של כל אחת מהן מופיע null byte כמקובל בשפת C.

n1 זה מספר התווים במחרוזת s1 כולל ה-null byte בסוף. למשל אם s1 זו המחרוזת "abc" אז n1 יהיה 4. בדומה לכך, n2 זה מספר התווים במחרוזת s2.

(בהינתן המחרוזות, ה-kernel יכול לחשב בעצמו את n1 ו-n2 אבל הם ניתנים כארגומנטים כדי לחסוך לו עבודה).

הארגומנט result הוא מצביע (לזיכרון הגלובלי של ה-device) שמשמש להחזרת התוצאה.

לצורך הפשטות, התכנית משגרת בלוק אחד בלבד של threads ונניח שמספר ה-threads בבלוק גדול או שווה ל-n1 וגם גדול או שווה ל-n2.

בתכנית מופיעה שורה

#define BLOCK_DIM 1024 המגדירה את מספר ה-threads בבלוק.

(במילים אחרות, BLOCK_DIM ו- BlockDim.x שווים).

ההשוואה בין שתי המחרוזות היא לפי "הסדר המילוני" (ידוע גם בשם "סדר לקסיקוגרפי").

אלגוריתם סדרתי של השוואה בין שתי מחרוזות s1 ו-s2 יעבוד כך: הוא ישווה בין התו הראשון של s1 לבין התו הראשון של s2, לאחר מכן ישווה בין התו השני של s1 לתו השני של s2 וכן הלאה. כשנתקלים בזוג תווים שאינם שווים אז מחזירים את ההפרש בין התו של s1 לבין התו של s2. (תווים הם מספרים (למשל קוד ascii) ולכן ניתן לקחת את ההפרש ביניהם). אם מגיעים לסוף של שתי המחרוזות בלי למצוא זוג תווים שאינם שווים אז מחזירים אפס.

דוגמאות:

המחרוזת ABC שווה למחרוזת ABC ולכן ההשוואה ביניהם תחזיר אפס
המחרוזת ABC גדולה מהמחרוזת AB: 'C' זה 67 (בקוד ascii). ה- null byte ערכו אפס
ולכן יוחזר $67 - 0 = 67$ 'C' - '\0'
המחרוזת CBAGG קטנה מהמחרוזת CBZ. 'A' זה 65, 'Z' זה 90 (בקוד ascii) ולכן ההשוואה תחזיר $65 - 90 = -25$ 'A' - 'Z'

הרעיון כאן הוא להיעזר בפעולת ה- scan כדי לערוך את ההשוואה באופן מקבילי.

קלט ופלט של התכנית

את שתי המחרוזות המשוויות ניתן לספק לתכנית כשני - command line arguments.
אם אין שני command line arguments אז התכנית תקרא את שתי המחרוזות מה- standard input.
המחרוזות מופרדות ע"י white space. כל זה כבר כתוב ואין צורך לגעת בקוד הזה.
הפלט (תוצאת ההשוואה בין המחרוזות) נכתב ל- standard output. גם זה כבר כתוב.

קימפול התכנית והרצתה

כדי לקמפל את הקובץ scan_strcmp.cu יש לתת את הפקודה הבאה
nvcc -gencode arch=compute_61,code=sm_61 scan_strcmp.cu -o foo

הקפידו לא להוסיף רווחים מיותרים (למשל סביב הפסיק). כאן -o foo אומר שקובץ ההרצה שנוצר יקרא foo (זו רק דוגמא – לא חייבים להשתמש בשם הזה). אם לא מספקים את האופציה -o אז קובץ ההרצה שנוצר יקרא a.out

ניתן להפעיל את התכנית כך:

./foo ABC ABZ

כאן סופקו שתי המחרוזות כ- `command line arguments`. לחילופין ניתן להכין קובץ (נקרא לו `myinput.txt`) עם שתי המחרוזות:

ABC

ABZ

ואז להפעיל את התכנית כך:

`./foo < myinput.txt`

התכנית קוראת מה- `standard input` שבתור ברירת מחדל זה לוח המקשים (ה- `terminal`) ולכן אפשר גם להפעיל את התכנית כך:

`./foo`

ולכתוב את שתי המחרוזות באופן אינטראקטיבי (בלוח המקשים)

הגשה:

התכנית שתגישו תהיה בקובץ שיקרא `scan_strcmp.cu` רק אחד מזוג שותפים יגיש במודל.

השם של קובץ ה- `zip` שתגישו (שיכלול רק את `scan_strcmp.cu` ואולי גם `makefile`) צריך לכלול את שמות שני המגישים (אם יש שניים—אפשר להגיש לבד).

צרפו גם קובץ `README` עם הפרטים של המגישים.

הערות

בתיקיית התרגולים של סמסטר קיץ 2023 יהיו בקרוב שני קבצים:

`scan_firstchar.cu` ו- `scan_unique.cu` בהם יש דוגמאות לשימוש ב- `scan`

(ב- `scan_firstchar.cu` מוצאים את המיקום של המופע הראשון של תו נתון במחרוזת נתונה. ב- `scan_unique.cu` הקלט הוא מערך של מספרים ממוינים. את המספרים מעתיקים למערך פלט תוך השמטה של מספרים "כפולים").

ה- `kernel` שעליכם לכתוב דומה במידה רבה ל- `kernels` שמופיעים בדוגמאות אלו.