



3^η Εργασία στο Μάθημα “Μικροεπεξεργαστές”

2023-2024

Δρ. Κεραμίδας Γεώργιος

Το project είναι ατομικό και θα πρέπει να υλοποιηθεί σε γλώσσα x86 assembly. Μπορείτε να υλοποιήσετε το project σε όποιον assembler θέλετε αλλά σας προτείνω να χρησιμοποιήσετε το MASM στο περιβάλλον Visual Studio (free version).

$$\begin{array}{l}
 \text{x direction} \\
 \begin{array}{|c|c|c|} \hline \text{green} & \text{green} & \text{green} \\ \hline \text{gray} & \text{white} & \text{dark blue} \\ \hline \text{blue} & \text{blue} & \text{blue} \\ \hline \end{array} \times \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} = \square \\
 \\
 \text{y direction} \\
 \begin{array}{|c|c|c|} \hline \text{green} & \text{green} & \text{green} \\ \hline \text{gray} & \text{white} & \text{dark blue} \\ \hline \text{blue} & \text{blue} & \text{blue} \\ \hline \end{array} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array} = \square
 \end{array}$$

Ο σκοπός του project είναι να υλοποιηθούν οι βασικοί kernels του αλγορίθμου ανίχνευσης ακμών Sobel σε x86 Assembly. Τα τμήματα διαχείρισης της εικόνας (διάβασμα αρχικού bmp αρχείου και αποθήκευση του τελικού αποτελέσματος θα γίνουν από την γλώσσα προγραμματισμού C). Το πέρασμα παραμέτρων μεταξύ C-Assembly θα πρέπει να γίνει μέσω του stack.

Σας δίνετε ένα έτοιμο πρόγραμμα σε C που υλοποιεί τον αλγόριθμο Sobel και sample input και output images. Στο αρχείο C υπάρχουν αρκετά σχόλια για να μπορέσετε

να καταλάβετε τα βήματα του αλγορίθμου. Συνολικά, υπάρχουν 3 συναρτήσεις της C που θα πρέπει να υλοποιηθούν σε x86 Assembly. Αυτές είναι:

Πρώτη συνάρτηση: Μετατροπή σε gray-scale images (1 μονάδα)

Converting the input RGB bmp to grayscale image (not black and white):

```
int bmptogray_conversion(int height,
                        int width,
                        RGBQUAD input_color[2048][2048],
                        int output_gray[2048][2048]);
```

Στη υλοποίηση της συγκεκριμένης συνάρτησης ΔΕΝ θα πρέπει να χρησιμοποιήσετε πράξεις κινητής υποδιαστολής. Σκεφτείτε κάποιο τρόπο να γίνει αυτό. Γενικά θα θέλαμε να πετύχουμε μια ακρίβεια 2 δεκαδικών ψηφίων.

Δεύτερη συνάρτηση: Υλοποίηση του αλγορίθμου Sobel (2 μονάδες)

Edge Detection with Sobel:

```
int sobel_detection(int height,
                  int width,
                  int input_gray_image[2048][2048],
                  unsigned char output_ee_image[2048][2048]);
```

Στη υλοποίηση της συγκεκριμένης συνάρτησης θα πρέπει να χρησιμοποιήσετε πράξεις κινητής υποδιαστολής (την μονάδα FPU).



Τρίτη συνάρτηση: Υπολογισμός των pixel στα άκρα της εικόνας (1 μονάδα)

Calculating the border pixels with replication:

```
int border_pixel_calculation(int height,  
                             int width,  
                             unsigned char ee_image[2048][2048]);
```

Στη υλοποίηση της συγκεκριμένης συνάρτησης θα πρέπει να χρησιμοποιήσετε εντολές τύπου SIMD (MMX, AVX).

Βήματα

Μελετήστε τον C κώδικα πριν προχωρήσετε στην υλοποίηση των αντίστοιχων kernels σε assembly. Το sensitivity του αλγορίθμου ορίζεται από ένα threshold το οποίο δηλώνεται στην αρχική του κώδικα σαν define.

Πληροφορίες για τους αλγορίθμους ανίχνευσης ακμών μπορείτε να βρείτε εδώ:

https://en.wikipedia.org/wiki/Edge_detection

Πληροφορίες για την μέθοδο Sobel μπορείτε να βρείτε εδώ:

https://en.wikipedia.org/wiki/Sobel_operator

Παραδοτέα

Τα παραδοτέα του project θα είναι ο κώδικας C-Assembly με επαρκή σχόλια (περίπου το 60% του κώδικα θα πρέπει να είναι σχόλια) καθώς και ένα output image του αλγορίθμου.

Ημερομηνία Παράδοσης

Η καταληκτική ημερομηνία παράδοσης του project είναι η 18^η Φεβρουαρίου 2024 (λήξη εξεταστικής).

Τρόπος Βαθμολόγησης

Το project δεν είναι υποχρεωτικό. Μπορείτε να κάνετε όσα μέρη της άσκησης θέλετε. Το πρώτο μέρος (συνάρτηση) αντιστοιχεί στο 10% της τελικής βαθμολογίας, το δεύτερο μέρος (συνάρτηση) αντιστοιχεί στο 20% και το τρίτο μέρος (συνάρτηση) αντιστοιχεί στο 10%.