

Αρχιτεκτονική Υπολογιστών
Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών
Εργασία Τρίτου Εξαμήνου

8 Φεβρουαρίου 2014

Μέλη της ομάδας

1. Ανέστης Δαλγκίτσης, 586, 3ο εξάμηνο
2. Γεώργιος Θεόδωρος Καλαμπόκης, 594, 3ο εξάμηνο
3. Χρήστος Παλαμιώτης, 648, 3ο εξάμηνο
4. Χρήστος Τόλης, 632, 3ο εξάμηνο

Εισαγωγή

- Το πρόγραμμα της ομάδας μας ελέγχει την εγκυρότητα πιστωτικών καρτών τύπου:
 - Mastercard
 - VISA
 - Amex
 - Diners Club
 - Discover
- Επίσης ως πρόσθετη λειτουργία έχει την δυνατότητα να δημιουργεί κάρτες των ίδιων τύπων ανάλογα με την προτίμηση του χρήστη.
- Η είσοδος για τον έλεγχο των καρτών γίνεται με ένα αρχείο txt που βρίσκεται στον ίδιο φάκελο με το εκτελέσιμο αρχείο

Δυνατότητες

- Ο χρήστης μπορεί να εισάγει όσες κάρτες επιθυμεί προς έλεγχο στο αρχείο txt, καθώς δεν υπάρχει περιορισμός.
- Υπάρχει έλεγχος για μη έγκυρους χαρακτήρες (όπως γράμματα και σύμβολα). Στην περίπτωση ύπαρξής τους, ο χρήστης ενημερώνεται με το κατάλληλο μήνυμα.

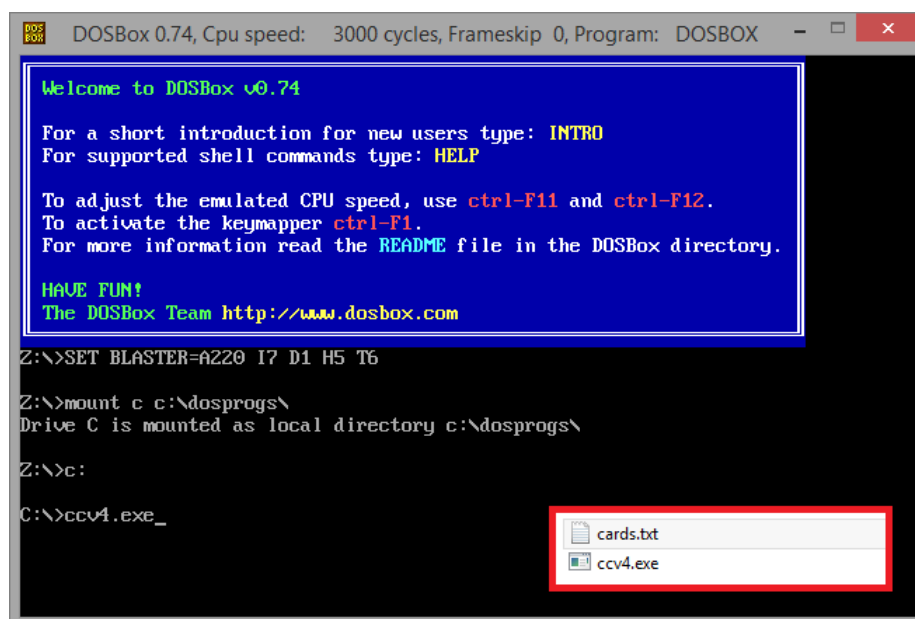
Απαιτήσεις για ορθή λειτουργία

- Το αρχείο txt πρέπει να βρίσκεται στον ίδιο φάκελο με το εκτελέσιμο αρχείο (ή στον φάκελο του δεβιγγερ αν εκτελεστεί προσομοίωση με το emu8086).
- Το αρχείο txt θα πρέπει να είναι κωδικοποίησης ANSI ή UTF-8 .
- Όλες οι κάρτες πρέπει να αναγράφονται **ΧΩΡΙΣ** κενά (πχ: 30569309025904) και να αναγράφονται **μία ανά γραμμή**.
- Ο μέγιστος αριθμός ψηφίων ανά κάρτα είναι 255 αριθμοί.

Οδηγός χρήσης

Πριν την εκτέλεση

1. Δημιουργούμε ένα αρχείο με όνομα cards.txt και το αποθηκεύουμε στον ίδιο φάκελο με το εκτελέσιμο αρχείο, σύμφωνα με τους περιορισμούς του προγράμματος.
2. Αναγράφουμε τους αριθμούς των πιστωτικών καρτών προς έλεγχο εγκυρότητας σύμφωνα με τους περιορισμούς του προγράμματος.



Εκτέλεση

1. Κατά την εκκίνηση του προγράμματος τυπώνεται στην οθόνη ένα σχέδιο κατασκευασμένο από χαρακτήρες ASCII που φανερώνει το όνομα του προγράμματός μας (CCV_4, δηλαδή Credit Card Validator version 4) καθώς και τα ονόματα των προγραμματιστών της ομάδας μας.
2. Για συνέχεια πατάμε τον χαρακτήρα ENTER (RETURN).



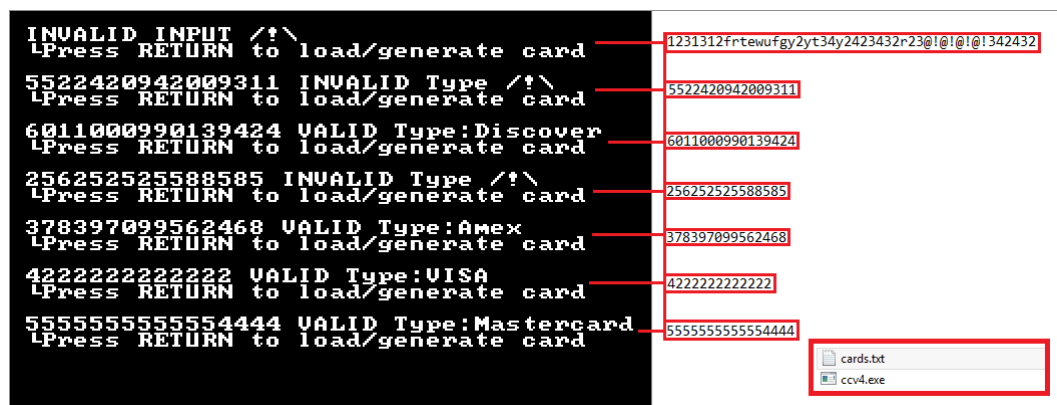
Κεντρικό μενού επιλογών

1. Στη συνέχεια εμφανίζεται ένα γραφικό περιβάλλον επιλογών.
2. Εδώ ο χρήστης καλείται να επιλέξει να επιλέξει ποια λειτουργία θέλει να εκτελέσει
 - (α') Πατώντας τον αριθμό 1, το πρόγραμμα μεταβαίνει στην οθόνη ελέγχου εγκυρότητας των πιστωτικών καρτών που αναγράφονται στο αρχείο cards.txt.
 - (β') Πατώντας τον αριθμό 2, το πρόγραμμα μεταβαίνει στην οθόνη δημιουργίας πιστωτικών καρτών.
 - (γ') Με τον αριθμό 3 τερματίζεται η εκτέλεση του προγράμματος
 - (δ') Ενώ με τον αριθμό 4 τυπώνεται η αρχική οθόνη.



Οθόνη ελέγχου εγκυρότητας πιστωτικών καρτών

1. Σε αυτή τη λειτουργία εμφανίζονται τα αποτελέσματα του ελέγχου εγκυρότητας των πιστωτικών καρτών που αναγράφονται στο αρχείο cards.txt.
 - (α') Τα αποτελέσματα εμφανίζονται με την σειρά που έχουν οι κάρτες.
 - (β') Στην περίπτωση έγκυρης κάρτας, αναγράφονται ο αριθμός της κάρτας καθώς και ο τύπος της.
 - (γ') Σε περίπτωση μη έγκυρης κάρτας, αναγράφεται ο αριθμός της καθώς και το αντίστοιχο μήνυμα μη-εγκυρότητας.
 - (δ') Σε περίπτωση μη αποδεκτών χαρακτήρων (δηλαδή συμβόλων, γραμμάτων και κενών χαρακτήρων), εμφανίζεται μόνο το μήνυμα μη έγκυρης εισόδου.
 - (ε') Η φόρτωση και ο έλεγχος νέας κάρτας γίνεται πατώντας το πλήκτρο ENTER (RETURN).
 - (ϛ') Η διαδικασία επαναλαμβάνεται μέχρι και τον έλεγχο της τελευταίας κάρτας.
 - (ζ') Αφού τελειώσει η διαδικασία, εμφανίζεται και πάλι το κεντρικό μενού επιλογών.



Οθόνη δημιουργίας πιστωτικών καρτών

- Σε αυτή τη λειτουργία ο χρήστης καλείται να επιλέξει τον τύπο και της/των πιστωτικής/πιστωτικών κάρτας/καρτών καθώς και τον αριθμό των καρτών που θέλει να δημιουργήσει.
 1. Αρχικά ο χρήστης επιλέγει τον τύπο των καρτών με τα πλήκτρα 0 έως 4, σύμφωνα με τις οδηγίες που δείχνει η οθόνη.
 2. Στην συνέχεια επιλέγει τον αριθμό των καρτών που θέλει να δημιουργήσει. **Προσοχή!** Ο αριθμός αυτός είναι τριψήφιος. Για παράδειγμα εάν ο χρήστης θέλει να δημιουργήσει 9 κάρτες θα πρέπει να γράψει 009.

```
[il Credit Card Generator <Beta v2>
Warning! Very slow process

[>] Type of card to generate:
  0 for Mastercard
  1 for VISA
  2 for Amex
  3 for Diners Club
  4 for Discover
[Type selected]: 0

[>] Number of cards to generate:
[Numbers Selected]: 003

5577777777777785  VALID Type:Mastercard
Press RETURN to load/generate card

5566666777777775  VALID Type:Mastercard
Press RETURN to load/generate card

5500000001111115  VALID Type:Mastercard
Press RETURN to load/generate card
```

- **Προσοχή!** Η δημιουργία έγκυρων πιστωτικών καρτών είναι πολύ χρονοβόρα και μπορεί να διαρκέσει από μερικά δευτερόλεπτα μέχρι μερικά λεπτά.
- Το αίτημα για δημιουργία και εμφάνιση νέας κάρτας γίνεται πατώντας το πλήκτρο ENTER (RETURN).
- Προτείνεται το αίτημα για νέα κάρτα να γίνεται μετά από λίγα δευτερόλεπτα από την εμφάνιση της προηγούμενης κάρτας
- Η διαδικασία επαναλαμβάνεται μέχρι και την δημιουργία της τελευταίας κάρτας.
- Αφού τελειώσει η διαδικασία, εμφανίζεται και πάλι το κεντρικό μενού επιλογών.

Περιγραφή Αλγορίθμων & Σημαντικά κομμάτια κώδικα

Το πρόγραμμα αυτό δημιουργήθηκε με βάση κάποιους κανόνες σχετικά με τον διαχωρισμό των λειτουργιών σε διάφορες συναρτήσεις και την επαναχρησιμοποίηση κώδικα

Η συνάρτηση readfile

- Ανοίγει μια ροή (stream) μόνο για την ανάγνωση (read-only) του αρχείου cards.txt ανά byte, με την βοήθεια της διακοπής INT 21h του λειτουργικού συστήματος DOS.
- Λειτουργεί με έναν κεντρικό βρόγχο επανάληψης readloop ο οποίος εκτελείται μέχρι και το τελευταίο byte του αρχείου.
 - Μέσα σε αυτόν τον βρόγχο εκτελούνται έλεγχοι εγκυρότητας εισόδου, οι μετατροπές των έγκυρων χαρακτήρων ASCII σε αριθμούς καθώς και η αποθήκευση αυτών στον πίνακα cardnum[255].
- Μόλις ολοκληρωθεί η ανάγνωση μιας έγκυρης γραμμής - κάρτας, προχωράμε στον έλεγχο του τύπου της κάρτας με την κλήση της συνάρτησης credit_card_type_finder και έπειτα στον έλεγχο της εγκυρότητας της κάρτας με την κλήση της συνάρτησης LUHNn (αλγόριθμος LUHN (mod 10)). Τέλος εκτελείται η τύπωση των αποτελεσμάτων από την συνάρτηση printcards.

Η συνάρτηση card_generator_menu

- Η card_generator_menu δέχεται από τον χρήστη τον τύπο καθώς και τον αριθμό των καρτών που θέλει να δημιουργήσει.
- Αφού ο χρήστης εισάγει τις προτιμήσεις του εκτελείται ένας βρόγχος επανάληψης ο οποίος για κάθε κάρτα καλεί τις κατάλληλες συναρτήσεις για την δημιουργία και την εμφάνιση των καρτών.
 - Μέσα σε αυτόν καλείται η συνάρτηση card_generatorη οποία επιλέγει το κατάλληλο πρόθεμα και μήκος ανάλογα με τον τύπο της κάρτας.
 - Οι υπόλοιποι αριθμοί της κάρτας επιλέγονται τυχαία από την συνάρτηση Pseudonoise_Number_Generator. Στην πραγματικότητα οι τιμές αυτές υπολογίζονται με βάση την ώρα και την ημερομηνία του συστήματος.
 - Αφού έχει δημιουργηθεί μια τυχαία κάρτα καλείται ο αλγόριθμος του LUHN για την επιβεβαίωση της έγκυρης τιμής. Αν ο αριθμός της τυχαίας κάρτας δεν είναι έγκυρος σύμφωνα με τον LUHN η διαδικασία επαναλαμβάνεται.
 - Μόλις ο LUHN εγκρίνει την τυχαία κάρτα καλείται η συνάρτηση printCards η οποία τυπώνει τον αριθμό και τον τύπο της.

Συναρτήσεις διαχείρισης γραφικών VGA

- Με την εκκίνηση του προγράμματος εκτελείται η συνάρτηση `setgraphics` που μεταβάλλει την ανάλυση της οθόνης για την σωστή τύπωση των γραφικών. Με την λήξη του προγράμματος κάθε γραφικό στοιχείο και χαρακτήρας ASCII απομακρύνεται με την συνάρτηση `optimiseScreen` και στην συνέχεια το σύστημα ανακτά την προεπιλεγμένη ανάλυση.
- Η συνάρτηση `splashscreen` δημιουργεί 2 γαλάζια πλαίσια στο πάνω και στο κάτω μέρος της οθόνης, γράφοντας στην μνήμη VGA τα αντίστοιχα pixels τους. Στη συνέχεια τυπώνεται το όνομα του προγράμματος με χαρακτήρες ASCII καθώς και τα στοιχεία των μελών της ομάδας μας.
- Η συνάρτηση `menu` πέραν από τις λειτουργίες επιλογής και ελέγχου, είναι εξ ολοκλήρου κατασκευασμένη με χαρακτήρες ASCII .

Δείκτες απόδοσης

Σημαντικά μεγέθη κώδικα

- Μέγεθος αρχείου πηγαίου κώδικα: 25 KB
- Μέγεθος αρχείου εκτελέσιμου αρχείου: 4 KB
- Μεγέθη κώδικα
 - Code Segment: 1566 bytes
 - Data Segment: 1434 bytes
 - Stack Segment: 256 bytes

Μη αποδοτικά κομμάτια κώδικα

1. Η συνάρτηση `cards_generator`

- Έχει κατασκευαστεί με τέτοιο τρόπο ώστε να καλείται διαρκώς η συνάρτηση που εκτελεί τον αλγόριθμο LUHN, μέχρι να βρεθεί μια τυχαία κάρτα που τον ικανοποιεί. Αυτό έχει ως αποτέλεσμα για την παραγωγή μιας κάρτας να χρειαστεί να κληθεί μέχρι και μερικές εκατοντάδες φορές καθιστώντας την όλη διαδικασία υπερβολικά αργή και χρονοβόρα.
- Το πρόβλημα της καθυστέρησης επιτείνεται όταν η ώρα δεν επιτρέπει την παραγωγή των επιθυμητών “τυχαίων” αριθμών. Αυτό συμβαίνει διότι οι τυχαίοι αριθμοί παράγονται με βάση τα δευτερόλεπτα της ώρας (wall clock time) καθώς δεν υπάρχει κάποια εντολή στον 8086 για την επιστροφή τυχαίων αριθμών. Στη συγκεκριμένη περίπτωση η διαδικασία καθυστερεί περισσότερο από ένα λεπτό και αφήνει τον χρήστη να πιστεύει πως η εφαρμογή δεν αποκρίνεται, κάτι που δεν ισχύει. Περισσότερες πληροφορίες σχετικά με το πρόβλημα αυτό θα βρείτε στην τελευταία σελίδα αυτού του εγγράφου.

2. Η συνάρτηση `Pseudonoise_Number_Generator`

- Η συνάρτηση αυτή αποτελεί το λιγότερο αποδοτικό κομμάτι του κώδικα καθώς καλείται το λιγότερο 13 φορές για να δημιουργήσει μια κάρτα. Τις περισσότερες φορές ο αλγόριθμος LUHN απορρίπτει την κάρτα που μόλις δημιουργήθηκε και έτσι επαναλαμβάνεται η όλη διαδικασία.
- Μέσα στην συνάρτηση αυτή υπάρχει ο βρόγχος pseudoLoop ο οποίος επαναλαμβάνεται κάθε φορά που ο τυχαίος αριθμός είναι μεγαλύτερος του 9. Στατιστικά επαναλαμβάνεται 2 από τις 6 φορές.
- Τέλος, το πέρασμα της τυχαίας τιμής γίνεται μέσω του τμήματος δεδομένων, μειώνει ελάχιστα την ταχύτητα εκτέλεσης.

Αξιοσημείωτες τεχνικές προγραμματισμού

- Χρησιμοποιήθηκαν καταχωρητές μέσα σε βρόγχους που εκτελούνται πάρα πολλές φορές.
- Οι περισσότερες μετακινήσεις δεδομένων μεταξύ μνήμης και καταχωρητών γίνονται πριν την είσοδο στους βρόγχους
- Συγκεκριμένοι πολλαπλασιασμοί και διαιρέσεις γίνονται με την λειτουργία shift για γρηγορότερα αποτελέσματα
- Η αρχικοποίηση των 16 bit καταχωρητών αντικαταστάθηκε με την εντολή XOR για ταχύτερη εκτέλεση και μικρότερη επιβάρυνση σε bytes .

Σχετικά με την διασωλήνωση

- Το πρόγραμμα αυτό έχει σχεδιαστεί για να εκτελείται στον επεξεργαστή 8086 της Intel ο οποίος δεν υποστηρίζει διασωλήνωση.
- Ενδεικτικά αναφέρονται κάποια παραδείγματα κινδύνων διασωλήνωσης
 1. RAW (Read After Write)
ADD CL,CH
MOV sum,CL
 2. WAR (Write After Read)
MOV BH,cardnum[DI]
MOV cardnum[SI],BH
 3. WAW (Write After Write)
MOV AL,cardnum[SI]
ADD AL,AL

Προβλήματα κατά την υλοποίηση και επίλυση αυτών

Πρόβλημα με τον έλεγχο εισόδου και η λύση του

Ένα σημαντικό πρόβλημα εμφανίστηκε κατά την δοκιμαστική εκτέλεση του προγράμματος, καθώς αυτό δεν διάβαζε την κάρτα που βρίσκονταν γραμμένη στην τελευταία γραμμή του αρχείου cards.txt. Το πρόβλημα αντιμετωπίστηκε ελέγχοντας αν το τελευταίο βψτε που διαβάστηκε είναι ο χαρακτήρας νεωλινε ή 0xA. Σε

αυτή την περίπτωση διαβάζεται ακόμη μια γραμμή, δηλαδή η τελευταία. Υπάρχει ξεχωριστός έλεγχος ώστε να μην διαβαστούν και άλλα bytes που δεν ανήκουν στο αρχείο.

Το πρόβλημα με την “γέννηση” πραγματικά τυχαίων αριθμών

Ένα πολύ σημαντικό πρόβλημα που παρουσιάστηκε είναι η παραγωγή πραγματικά τυχαίων αριθμών. Μετά από αρκετή ώρα αναζήτησης στο internet όπως και στις διαφάνειες του μαθήματος, διαπιστώσαμε πως όσοι αλγόριθμοι ψευδοτυχαίων αριθμών δοκιμάσαμε μας έδιναν συνεχώς τον ίδιο αριθμό και συνεπώς, τις ίδιες κάρτες. Αυτό συμβαίνει γιατί όλοι αυτοί οι αλγόριθμοι χρησιμοποιούν το ρολόι του συστήματος (wall clock time) για να πάρουν μια τιμή, να την επεξεργαστούν και στη συνέχεια να την επιστρέψουν στο πρόγραμμα. Όμως αυτή η συνάρτηση καλείται εκατοντάδες φορές το δευτερόλεπτο και δεν προλαβαίνει να ανανεωθεί. Δοκιμάσαμε να επιστρέφουμε τιμή μόνο σε περίπτωση που αυτή δεν είναι ίδια με την προηγούμενη, αλλά αυτό δεν αποτελεί λύση καθώς καθυστερεί σημαντικά την όλη διαδικασία (10 λεπτά για μια κάρτα). Για αυτό το λόγο και απορρίφθηκε η επιδιόρθωση αυτού το προβλήματος. Παρόμοιο πρόβλημα με την παραγωγή τυχαίων αριθμών είχε παρουσιαστεί και στην εργαστηριακή άσκηση 8.