

ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ (2005-06)

Άσκηση 6

Υλοποιήστε σε C τα προγράμματα `pgmread` και `pgmwrite` που συναντήσατε στην Άσκηση 3. Τις προδιαγραφές χρήσης των προγραμμάτων αυτών τις γνωρίζετε ήδη. Δείτε πάλι ένα παράδειγμα εκτέλεσης του `pgmread`:

```
% ./pgmread demo.pgm
8 16
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 150 150 150 150 150 150 255 255 255 255 255
255 255 150 150 150 150 255 255 255 255 150 150 150 150 255 255
255 255 182 182 150 255 0 0 0 0 255 150 182 182 255 255
255 255 182 182 150 255 0 0 0 0 255 150 182 182 255 255
255 255 150 150 150 150 255 255 255 255 150 150 150 150 255 255
255 255 255 255 255 150 150 150 150 150 150 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
%
```

Σε κάθε περίπτωση όμως, διαβάστε πάλι την εκφώνηση της Άσκησης 3, τουλάχιστον το τμήμα της που δείχνει τη χρήση των `pgmread` και `pgmwrite`.

Αυτό που χρειάζεται, επιπλέον αυτών που γνωρίζετε ήδη, για να μπορέσετε να γράψετε τα δύο προγράμματα `pgmread.c` και `pgmwrite.c` είναι η δομή ενός αρχείου σε `pgm format`. Η περιγραφή της δομής αυτής ακολουθεί.¹ Ένα αρχείο σε `pgm format` περιλαμβάνει, κατά σειρά, τα εξής:

- Το “μαγικό αριθμό” που χαρακτηρίζει τον τύπο του αρχείου. Για αρχεία σε `pgm format`, ο “μαγικός αριθμός” είναι οι δύο χαρακτήρες “P5”.
- Λευκό διάστημα, δηλαδή έναν ή περισσότερους λευκούς χαρακτήρες (κενά, στηλογνώμονες, αλλαγές γραμμής).
- Το πλάτος της εικόνας, εκφρασμένο με τους ASCII κωδικούς των ψηφίων του στο δεκαδικό σύστημα. Για παράδειγμα, αν το πλάτος είναι 16, μέσα στο αρχείο, θα υπάρχουν στο σημείο αυτό τα ‘1’ και ‘6’ (για την ακρίβεια, οι τιμές 49 και 54 που είναι οι ASCII κωδικοί των ψηφίων αυτών, τους οποίους όμως δεν χρειάζεται να γνωρίζετε για την υλοποίηση της άσκησης).
- Λευκό διάστημα.
- Το ύψος της εικόνας, εκφρασμένο και αυτό με τους ASCII κωδικούς των ψηφίων του στο δεκαδικό σύστημα, όπως, δηλαδή, και το πλάτος.
- Λευκό διάστημα.

¹Για την ακρίβεια, η δομή που περιγράφεται είναι ένα υποσύνολο της “επίσημης” δομής ενός αρχείου σε `pgm format`. Υπάρχουν και κάποιες, δευτερεύουσες μάλλον, λεπτομέρειες, οι οποίες δεν περιγράφονται και, φυσικά, δεν σας ζητείται να υλοποιήσετε. Πάντως, τα αρχεία σε `pgm format` που είχαν δοθεί για πειραματισμούς στα πλαίσια της Άσκησης 3 καλύπτονται από τη δομή που σας δίνεται. Υπάρχει όμως το ενδεχόμενο κάποιο άλλο αρχείο που θα βρείτε στο διαδίκτυο ή θα παραγάγετε από κάποιο πρόγραμμα, να μην καλύπτεται από τη δομή αυτή, οπότε το πρόγραμμα `pgmread` που θα γράψετε να μην το αναγνωρίζει σωστά, όπως, άλλωστε, ούτε και το `pgmread` που σας δόθηκε για την Άσκηση 3, παρότι θα είναι πιθανώς ένα καθ’ όλα νόμιμο αρχείο σε `pgm format`.

- Τη μέγιστη τιμή που μπορεί να έχει ένα εικονοστοιχείο μέσα στην εικόνα. Η τιμή για κάθε εικονοστοιχείο εκφράζει την απόχρωση του γκρι για το εικονοστοιχείο. Η μέγιστη τιμή απόχρωσης του γκρι, που βρίσκεται στο αρχείο στο σημείο αυτό, αναπαρίσταται επίσης με τους ASCII κωδικούς των ψηφίων της στο δεκαδικό σύστημα. Συνήθως, η τιμή αυτή είναι 255. Θεωρήστε ότι για τις ανάγκες της άσκησης, η τιμή αυτή είναι πάντα 255, οπότε στη θέση αυτή στο αρχείο θα υπάρχουν τα '2', '5' και '5' κατά σειρά (δηλαδή οι τιμές 50, 53 και 53).
- Ένας μοναδικός λευκός χαρακτήρας.
- Μία ακολουθία από (πλάτος × ύψος) bytes που αντιπροσωπεύουν τις τιμές απόχρωσης του γκρι για τα εικονοστοιχεία της εικόνας, με διάταξη κατά γραμμές.
- Μετά από το λευκό διάστημα που ακολουθεί τον μαγικό αριθμό (P5) και πριν από την πληροφορία για τη μέγιστη τιμή απόχρωσης του γκρι, το αρχείο μπορεί να περιλαμβάνει και σχόλια. Ένα σχόλιο εκτείνεται από ένα χαρακτήρα '#' έως την επόμενη αλλαγή γραμμής ('\n'). Μπορεί να υπάρχουν σχόλια μετά από το λευκό διάστημα που ακολουθεί την πληροφορία του πλάτους ή του ύψους της εικόνας, στην ίδια γραμμή, αλλά μπορεί να υπάρχουν και γραμμές που να περιέχουν μόνο σχόλια. Φυσικά, το πρόγραμμα `pgmread` που θα γράψετε, θα πρέπει να αγνοεί σχόλια που βρίσκει στο αρχείο που διαβάζει. Όσον αφορά το `pgmwrite` πρόγραμμα που θα γράψετε, φροντίστε μέσα στα αρχεία που θα δημιουργείτε να βάζετε τουλάχιστον ένα σχόλιο που να δείχνει την ταυτότητά σας, δηλαδή του δημιουργού του προγράμματος που κατασκευάζει αρχεία σε `pgm format`.

Αν θέλετε να δείτε το “εσωτερικό” ενός αρχείου σε `pgm format`, χρησιμοποιήστε την εντολή του Unix “`od -tu1`” (“`man od`” για λεπτομέρειες). Δείτε ένα παράδειγμα:

```
% od -tu1 demo.pgm
0000000 080 053 010 049 054 032 056 010 050 053 053 010 255 255 255 255
0000020 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
0000040 255 150 150 150 150 150 150 255 255 255 255 255 255 150 150
0000060 150 150 255 255 255 255 150 150 150 150 255 255 255 255 182 182
0000100 150 255 000 000 000 000 255 150 182 182 255 255 255 255 182 182
0000120 150 255 000 000 000 000 255 150 182 182 255 255 255 255 150 150
0000140 150 150 255 255 255 255 150 150 150 150 255 255 255 255 255 255
0000160 255 150 150 150 150 150 150 255 255 255 255 255 255 255 255
0000200 255 255 255 255 255 255 255 255 255 255 255 255
0000214
%
```

Κάθε γραμμή που μας τυπώνει η εντολή `od` στο παραπάνω παράδειγμα περιέχει 16 bytes του αρχείου, όπου η θέση του πρώτου απ' αυτά μέσα στο αρχείο φαίνεται στην πρώτη στήλη αριστερά (σε οκταδική μορφή). Τα 080 και 053 στην αρχή είναι οι ASCII κωδικοί των χαρακτήρων του μαγικού αριθμού (P5). Μετά υπάρχει μία αλλαγή γραμμής (ASCII κωδικός 010), το πλάτος της εικόνας, που είναι 16 (ASCII κωδικοί 049 και 054 των χαρακτήρων '1' και '6'), ένας κενός χαρακτήρας (ASCII κωδικός 032), το ύψος της εικόνας, που είναι 8 (ASCII κωδικός 056 για τον χαρακτήρα '8'),² μία αλλαγή γραμμής, η μέγιστη τιμή απόχρωσης του γκρι 255 (ASCII κωδικοί 050, 053 και 053 των '2', '5' και '5', αντίστοιχα) και, τέλος, μία αλλαγή γραμμής. Μετά, ακολουθούν 16×8 (=128) bytes με τις τιμές των εικονοστοιχείων της εικόνας.

Δείτε και σε μορφή χαρακτήρων τα πρώτα 12 bytes του αρχείου `demo.pgm`:

²Προσέξτε ότι ενώ μέσα στο αρχείο πρώτα βρίσκεται το πλάτος της εικόνας και μετά το ύψος, το πρόγραμμα `pgmread` εκτυπώνει στην πρώτη γραμμή της εξόδου πρώτα το ύψος και μετά το πλάτος.

```
% od -tu1c -N 12 demo.pgm
0000000 080 053 010 049 054 032 056 010 050 053 053 010
          P 5 \n 1 6      8 \n 2 5 5 \n
0000014
%
```

Επίσης, παρατηρήστε και το εσωτερικό αρχείων σε pgm format με σχόλια:

```
% od -tu1c -N 60 cartman.pgm
0000000 080 053 010 035 032 067 082 069 065 084 079 082 058 032 084 104
          P 5 \n #      C R E A T O R :      T h
0000020 101 032 071 073 077 080 039 115 032 080 078 077 032 070 105 108
          e      G I M P ' s      P N M      F i l
0000040 116 101 114 032 086 101 114 115 105 111 110 032 049 046 048 010
          t e r      V e r s i o n      1 . 0 \n
0000060 049 053 048 032 049 052 053 010 050 053 053 010
          1 5 0      1 4 5 \n 2 5 5 \n
0000074
%
```

```
% od -tu1c -N 58 stsmall.pgm
0000000 080 053 010 035 032 067 082 069 065 084 079 082 058 032 088 086
          P 5 \n #      C R E A T O R :      X V
0000020 032 086 101 114 115 105 111 110 032 051 046 049 048 097 032 032
          V e r s i o n      3 . 1 0 a
0000040 082 101 118 058 032 049 050 047 050 057 047 057 052 010 051 050
          R e v :      1 2 / 2 9 / 9 4 \n 3 2
0000060 048 032 050 052 048 010 050 053 053 010
          0      2 4 0 \n 2 5 5 \n
0000072
%
```

Για να δείτε “εσωτερικά” ένα αρχείο σε pgm format στα Windows, κατεβάστε το πρόγραμμα που θα βρείτε εδώ:

<http://www.hhdsoftware.com/free-hex-editor.html>

Για την άσκηση αυτή, θα πρέπει να παραδώσετε ένα zip αρχείο που θα περιλαμβάνει τα δύο πηγαία προγράμματα pgmread.c και pgmwrite.c. Οδηγίες για τη δημιουργία zip αρχείων υπάρχουν στην εκφώνηση της Άσκησης 5.

Οι οδηγίες υποβολής της άσκησης αυτής, αλλά και των Ασκήσεων 4 και 5, έχουν ανακοινωθεί στη λίστα και στην ιστοσελίδα του μαθήματος.