Προγραμματισμός Ι

Ψευδοτυχαίοι Αριθμοί

Δημήτρης Μιχαήλ

Τμήμα Πληροφορικής και Τηλεματικής
Χαροκόπειο Πανεπιστήμιο

Ψευδοτυχαίοι Αριθμοί

Μια γεννήτρια ψευδοτυχαίων αριθμών είναι ένας αλγόριθμος παραγωγής μιας σειράς αριθμών που προσεγγίζουν τις ιδιότητες των τυχαίων αριθμών.

Η C μας παρέχει στην βιβλιοθήκη της μια τέτοια γεννήτρια.

Η συνάρτηση `rand()`

- Η συνάρτηση **`rand`**() δηλώνεται στο αρχείο **`stdlib.h`** και μας παρέχει την δυνατότητα παραγωγής ψευδοτυχαίων αριθμών.

- Η συνάρτηση επιστρέφει έναν ακέραιο στο διάστημα $[0, \mathit{RAND\_MAX}]$ όπου **`RAND_MAX`** είναι μια σταθερά.

- Με βάση το πρότυπο της C η σταθερά **`RAND_MAX`** πρέπει να έχει τουλάχιστον την τιμή 32767. Συνήθως όμως έχει μεγαλύτερη τιμή. Στο σύστημα του ομιλητή η **`RAND_MAX`** έχει την τιμή 2147483647.

Η συνάρτηση `rand()`

- Εάν οι αριθμοί ήταν πραγματικά τυχαίοι τότε κάθε φορά που θα καλούσαμε την **`rand`**() θα περιμέναμε να πάρουμε κάποιον αριθμό με πιθανότητα

$$\frac{1}{1 + \text{RAND\_MAX}}$$

- Επειδή όμως η γεννήτρια παράγει ψευδοτυχαίους αριθμούς η πιθανότητα δεν είναι η παραπάνω.

- Για να καταλάβουμε όμως διαφορά πρέπει να χρησιμοποιήσουμε την γεννήτρια πάρα πολλές φορές συνεχόμενα.

Παράδειγμα

Η συνάρτηση **rand**()

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      int i;
6      for(i = 0; i < 10; i++)
7          printf("%d ", rand());
8
9      return 0;
10 }
```

Το παραπάνω πρόγραμμα στο σύστημα του ομιλητή εκτυπώνει:

```
1804289383 846930886 1681692777 1714636915
1957747793 424238335 719885386 1649760492
596516649 1189641421
```

Μετατροπή Διαστήματος

Η συνάρτηση **rand**()

Για να μετατρέψουμε το διάστημα

- από 0 έως **RAND_MAX**
- σε 0 έως N όπου $N < RAND\_MAX$

χρησιμοποιήσουμε τον τελεστή υπολοίπου διαίρεσης.

Διαιρούμε δηλαδή τον αριθμό που μας δίνει η **rand**() με $N + 1$.

π.χ

```
int r = rand() % 100;
```

Παραγωγή Ψευδοτυχαίων από 0 έως 99

```c
#include <stdio.h>
#include <stdlib.h>

int main() {
    int i;
    for(i = 0; i < 10; i++)
        printf("%d ", rand()%100);
    printf("\n");

    return 0;
}
```

Το παραπάνω πρόγραμμα στο σύστημα του ομιλητή εκτυπώνει:

```
83 86 77 15 93 35 86 92 49 21
```

Πολλαπλή Εκτέλεση

Παραγωγή Ψευδοτυχαίων από 0 έως 99

```c
#include <stdio.h>
#include <stdlib.h>

int main() {
    int i;
    for(i = 0; i < 10; i++)
        printf("%d ", rand()%100);
    printf("\n");

    return 0;
}
```

Άμα εκτελέσουμε το παραπάνω πρόγραμμα 2 φορές θα εκτυπώσει:

```
83 86 77 15 93 35 86 92 49 21
83 86 77 15 93 35 86 92 49 21
```

Τρόπος Λειτουργίας Γεννήτριας

- Η γεννήτρια ξεκινάει πάντα από τον αριθμό 1 και κάνοντας διάφορους μετασχηματισμούς παράγει την σειρά των αριθμών.
- Ο αριθμός που ξεκινάει η γεννήτρια λέγεται **σπόρος** (seed) της γεννήτριας.
- Η αρχική τιμή του σπόρου είναι πάντα 1 και άρα όσες φορές και να εκτελέσουμε το πρόγραμμα μας θα εκτυπώνει τις ίδιες τιμές.

Αλλαγή του Σπόρου

Η συνάρτηση **`srand`**()

Η C μας παρέχει την συνάρτηση **`srand`**() ώστε να μπορούμε να αλλάξουμε τον σπόρο. Η συνάρτηση **`srand`**() πέρνει ως παράμετρο έναν **`unsigned int`**.

Αφού καλέσουμε την **`srand`**() η γεννήτρια παράγει μια καινούρια σειρά ψευδοτυχαίων αριθμών.

Άλλη Σειρά

```c
#include <stdio.h>
#include <stdlib.h>

int main() {
    int i;

    srand(2);

    for(i = 0; i < 10; i++)
        printf("%d ", rand()%100);
    printf("\n");

    return 0;
}
```

Άμα εκτελέσουμε το παραπάνω πρόγραμμα θα εκτυπώσει:

```
90 19 88 75 61 98 64 77 45 27
```

Κοινά Λάθη

```c
#include <stdio.h>
#include <stdlib.h>

int main() {
    int i;
    srand(2);
    for(i = 0; i < 10; i++)
        printf("%d ", rand()%100);
    printf("\n");

    srand(3); // No! Never initialize twice

    for(i = 0; i < 10; i++)
        printf("%d ", rand()%100);
    printf("\n");

    return 0;
}
```

Η δεύτερη αρχικοποίηση της γεννήτριας χαλάει όλες τις καλές ιδιότητες που έχει.

Κορώνα ή Γράμματα

heads or tail

```c
#include <stdio.h>
#include <stdlib.h>

#define HEADS 0
#define TAILS 1

int flip() {
    return rand()%2;
}

int main() {
    int i;
    for(i = 0; i < 20; i++)
        printf("%c ", flip()==HEADS?'H':'T');

    return 0;
}
```

Στον υπολογιστή του ομιλητή το παραπάνω πρόγραμμα εκτυπώνει:

```
H T H H H H T T H H T H T H H T T T T T
```

Σπόρος και Χρόνος Συστήματος

Οι προγραμματιστές όταν χρησιμοποιούν την γεννήτρια ψευδοτυχαίων αριθμών χρησιμοποιούν πολλές φορές τον χρόνο του συστήματος ως σπόρο. Έτσι είναι σχεδόν απίθανο να χρησιμοποιούν τον ίδιο σπόρο συνέχεια.

Η C μας παρέχει το αρχείο

```
#include <time.h>
```

που περιέχει συναρτήσεις και τύπους για να διαβάσουμε τον χρόνο του συστήματος.

Σπόρος και Χρόνος Συστήματος

Οι προγραμματιστές όταν χρησιμοποιούν την γεννήτρια ψευδοτυχαίων αριθμών χρησιμοποιούν πολλές φορές τον χρόνο του συστήματος ως σπόρο. Έτσι είναι σχεδόν απίθανο να χρησιμοποιούν τον ίδιο σπόρο συνέχεια.

Η C μας παρέχει το αρχείο

```
#include <time.h>
```

που περιέχει συναρτήσεις και τύπους για να διαβάσουμε τον χρόνο του συστήματος.

Προσοχή ώστε να μην αρχικοποιούμε την γεννήτρια παραπάνω από μία φορές, αλλιώς χαλάνε οι ιδιότητες της όπως το μήκος περιόδου.

Η συνάρτηση `time()`

Η συνάρτηση **`time`**() μας επιστρέφει τον χρόνο που έχει περάσει από την χρονική στιγμή **00:00:00 UTC, January 1, 1970**, σε δευτερόλεπτα.

```
1  #include <stdio.h>
2  #include <time.h>
3
4  int main() {
5      printf("%ld", time(NULL));
6
7      return 0;
8  }
```

Το παραπάνω πρόγραμμα εκτυπώνει την ώρα που εκτελείται το πρόγραμμα, π.χ η διαφάνεια αυτή πρωτοεμφανίστηκε την χρονική στιγμή:

```
1259243300
```

Χρόνος και Σπόρος

Κορώνα ή Γράμματα

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define HEADS 0
#define TAILS 1

int flip() {
    return rand()%2;
}

int main() {
    int i;
    srand(time(NULL));
    for(i = 0; i < 20; i++)
        printf("%c ", flip()==HEADS?'H':'T');
    return 0;
}
```

Πλέον είναι πολύ σπάνιο να εκτυπώσει την ίδια σειρά παραπάνω από μια φορά.

Μία Απλή Γεννήτρια

Linear Congruential Generator

Στην πραγματική ζωή πάρα πολύ σπάνια θα χρειαστεί να γράψει κάποιος μία γεννήτρια.

Ας δούμε όμως μία απλή τεχνική.

Μία Απλή Γεννήτρια

Linear Congruential Generator

```
1  #define a (1103515245)
2  #define c (12345)
3  #define m (1<<31)
4
5  static unsigned int seed = 1;
6
7  void srand(unsigned int s) {
8      seed = s;
9  }
10
11 int rand() {
12     seed = (a * seed + c) % m;
13     return seed;
14 }
```

Μία Απλή Γεννήτρια

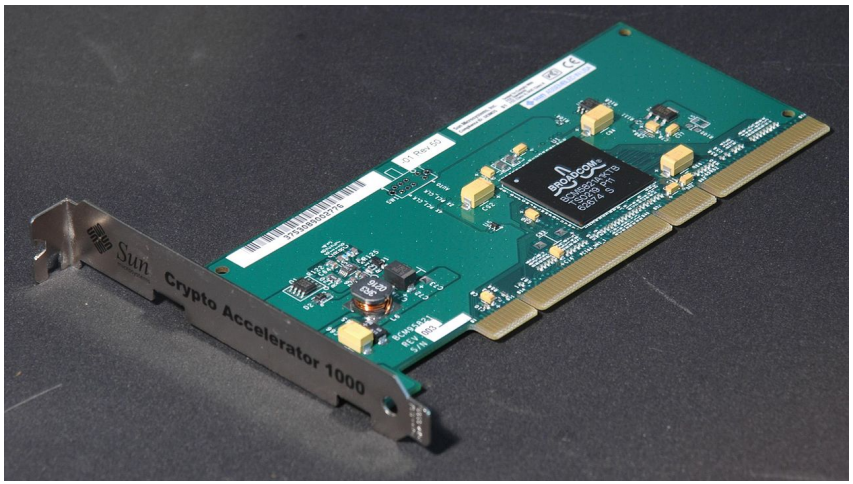
Linear Congruential Generator

Ο αλγόριθμος λειτουργεί καλά μόνο για συγκεκριμένες τιμές των σταθερών a, c και m.

Για περισσότερες λεπτομέρειες μπορείτε να διαβάσετε το παρακάτω άρθρο στην wikipedia: `https://en.wikipedia.org/wiki/Linear_congruential_generator`

Γεννήτρια Τυχαίων Αριθμών (TRNG)

Μπορούμε να φτιάξουμε γεννήτρια που να παράγει πραγματικά τυχαίους αριθμούς; Ναι, αλλά χρειαζόμαστε υλικό (hardware).



Εικόνα από Retro-Computing Society of Rhode Island

Για περισσότερες πληροφορίες https://en.wikipedia.org/wiki/Hardware_random_number_generator.

Μελέτη

- Κεφάλαιο 5.9, C Προγραμματισμός, Deitel & Deitel, 3η έκδοση
- Donald Knuth, Art Of Computer Programming, Seminumerical Algorithms, Third Edition, Addison-Wesley, 1997.
- Μία γρήγορη γεννήτρια παραγωγής ψευδοτυχαίων αριθμών (Mersenne Twister). `http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html`
- `http://www.gnu.org/s/gsl/manual/html_node/Random-Number-Generation.html`