Hashing - Funcții de repartizare

Universitatea ”Transilvania” din Brașov

May 16, 2018

Repartizarea cheilor de tip șir de caractere

Additive hash:

```
unsigned int add_hash(char *key)
{
    unsigned int len = strlen(key);
    unsigned int h = 0;
    int i;

    for (i = 0; i < len; i++)
    {
        h += key[i];
    }

    return h;
}
```

Repartizarea cheilor de tip șir de caractere

XOR hash:

```
unsigned int xor_hash(char *key)
{
    unsigned int len = strlen(key);
    unsigned int h = 0;
    int i;

    for (i = 0; i < len; i++)
    {
        h^ = key[i];
    }

    return h;
}
```

Repartizarea cheilor de tip șir de caractere

Rotational hash:

```
unsigned int rot_hash(char *key)
{
    unsigned int len = strlen(key);
    unsigned int h = 0;
    int i;

    for (i = 0; i < len; i++)
    {
        h = (h << 5) ^ (h >> 27) ^ key[i];
    }


    return h;
}
```

Repartizarea cheilor de tip șir de caractere

Bernstein hash:

```
unsigned int djb_hash(char *key)
{
    unsigned int len = strlen(key);
    unsigned int h = 0;
    int i;

    for (i = 0; i < len; i++)
    {
        h = 33 * h + key[i];
    }

    return h;
}
```

Repartizarea cheilor de tip șir de caractere

FVN hash: Algoritm general

```
hash = FNV_offset_basis
for each byte_of_data to be hashed
          hash = hash $\times$ FNV_prime
          hash = hash XOR byte_of_data
return hash
```

Repartizarea cheilor de tip șir de caractere

FVN hash - pentru 32 bits

```
unsigned int djb_hash(char *key)
{
    unsigned int len = strlen(key);
    unsigned int h = 2166136261;
    int i;

    for (i = 0; i < len; i++)
    {
        h = (h * 16777619) ^ key[i];
    }

    return h;
}
```