

Proiect Sisteme de Operare

Moarcas Cosmin-Ionut

Cerinta

Sa se implementeze un encriptor/decriptor care primeste un fisier de intrare cu diferite cuvinte. Programul mapeaza fisierul de intrare in memorie si porneste mai multe procese care vor apela o permutare random pentru fiecare cuvint. Permutarile vor fi scrise intr-un fisier de iesire. Programul poate primi ca argument doar fisierul de intrare, in acest caz va face criptarea cuvintelor; sau va primi fisierul avand cuvintele criptate si permutarile folosite pentru criptare, caz in care va genera fisierul de output avand cuvintele decriptate.

Voi alege o permutare, fiecare proces va avea de prelucrat o parte din fisier, aplicand acea permutare pe partea lui de fisier.

Maparea fisierului in memorie

Pentru a mapa fisierul in memorie am avut de ales intre doua variante:

Varianta I: Inainte de a crea procesele, mapez tot fisierul de intrare in memorie si creez n procese, unde n reprezinta numarul de cuvinte din fisier.

Avantaj: Fac un singur apel in care mapez fisierul, ceea ce creste performanta programului.

Dezavantaj: In cazul in care primesc un fisier de intrare de dimensiuni mari, pot avea erori.

Varianta II: Creez procesele, iar in fiecare proces mapez cate o parte a fisierului de dimensiunea unei pagini.

Avantaj: Nu vor aparea erori, chiar daca dimensiunea fisierului este foarte mare, deoarece mapez mereu parti de dimensiunea unei pagini.

Dezavantaj: Mai lent, deoarece se va mapa de mai multe ori fisierul. De asemenea aceasta varianta este mai greu de implementat.

În acest proiect am ales varianta I. Marele dezavantaj pe care l-am observat în cea de-a doua variantă este faptul că fiecare parte de fișier ar avea aceeași cheie/permutare. Asta înseamnă că va trebui să criptez cuvinte de lungimi diferite cu o cheie fixă. Acest lucru se poate realiza în următorii pași:

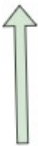
- aleg o cheie de dimensiune mare
- dacă un cuvânt este mai scurt decât cheia, adaug la sfârșitul cuvântului simbolul #

Criptarea


Ideea:

Pentru a cripta cuvintele a fost nevoie mai întâi să aflăm poziția fiecăruia în memoria mapată. Într-un vector de perechi am stocat pentru fiecare cuvânt în parte poziția și lungimea acestuia în fișierul mapat.

C	r	y	p	t		d	e	c	y	p	t	\0
0	1	2	3	4	4	5	6	7	7	8	9	10



index cuvânt = 0



index cuvânt = 2

```
pozitii[index_cuvant = 0].pozitie_start = 0
pozitii[index_cuvant = 0].lungime = 5
```

```
pozitii[index_cuvant = 1].pozitie_start = 5
pozitii[index_cuvant = 1].lungime = 6
```

Fiecare proces se va ocupa de un singur cuvânt. Procesul va copia cuvântul din fisier într-o variabilă separată și va apela o funcție de criptare care va modifica variabila trimisă ca parametru. După ce se efectuează criptarea, voi copia variabila înapoi în fisierul mapat.

Criptarea cuvântului

Criptarea cuvântului se realizează într-o funcție separată. Mai întâi generez o permutare/cheie (o rearanjare aleatoare a pozițiilor cuvântului) iar apoi schimb ordinea literelor din cuvânt cu ajutorul cheii. În final, salvez cheia împreună cu indexul cuvântului într-un fisier de permutări. Pentru a sincroniza scrierea în fisier, am utilizat un mutex. Înainte ca un proces să înceapă să scrie în fisier, mutex-ul se va bloca, iar când procesul termină de scris, mutex-ul se va debloca.

De asemenea, acest fisier de permutări a fost mapat în memorie, altfel descriptorul fisierului nu ar fi fost comun tuturor proceselor.

Decriptarea

Pentru a decripta cuvintele trebuie mai întâi să salvez într-o structură de date cheile. Am utilizat o matrice care pentru fiecare index al unui cuvânt stoca cheia asociată acestuia.

Metoda de decriptare este aproape identică cu cea de criptare a cuvintelor.

Git proiect: <https://github.com/Moarcas/Encriptor.git>