

Multiprocesorski sistemi – Zadatak 1 – Merge Sort

Ovaj document ukratko opisuje nacin rada algoritma za Merge Sort. Koriscen je kod preuzet sa interneta za sam algoritam Merge sortiranja I time je “kreiran” sekvencijalni program uz dodato merenje vremena I randomizovani niz.

Paralelan kod. pored funkcija mergeSort I merge koristi I dodatne :

1. checkSort – Proverava da li je niz dobro sortiran
2. printVector – Ispis vektora
3. sum – Suma prvih k clanova vektora
4. divide – Rekurzivna funkcija koja za datu velicinu niza vraca velicine kreiranih podnizova u stablolikom maniru

U main kodu se radi upis u fajlove, randomizacija niza I startovanje merenja vremena. Na osnovu datog broja niti se proracunava do kojeg nivoa niz mora da se deli kako bi niti bili efikasno zaposlene. Pomocu divide funkcije dobijamo velicine podnizova koje koristimo za lakse indeksiranje lokalnih starova I endova za svaku pojedinačnu nit. Svaka nit poziva mergeSort nad svojim pod nizom I dobija se delimicno sortiran niz.

Recimo od niza 7 4 3 4 5 2 8 1 za 4 niti dobijamo :

4 7 3 4 2 5 1 8 (svaka cetvrtina je sortirana)

Nakon toga je potrebno vratiti se do vrha stabla a to se radi merge funkcijom. I ovde je radjeno “manevrisanje” pomocu posebnog subSize vektora koji sadrzi pomenute velicine podnizova, sa tim da je on promenljiv u toku koda kako bi algoritam radio u opstem slucaju.(iako ovo dodaje tehnicki na kasnjenju merenjem je ustanovljeno da je to zanemarljivo).

Number of threads – 10M numbers	Execution time	Speed-Up
Sequential(Single) thread		-
Parallel code – 2 threads		
Parallel code – 3 threads		

Parallel code – 4 threads		
---------------------------	--	--