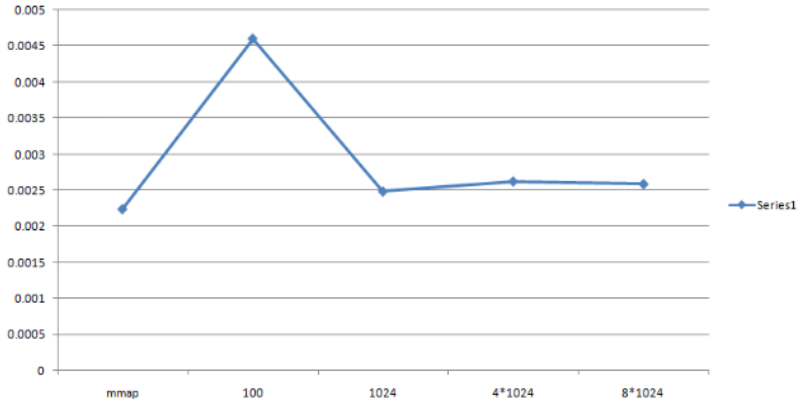
بنام خدا

میلاد تیموری 915222059

سوال 1) در این پروژه برای هر کدام از تردهای خواندن،نوشتن،جمع،منها،تقسیم،ضرب یک بافر 5تایی از جنس struct به علاوه یک mutex و 2 سمافور full و empty با مقادیر اولیه 0 و 5 در نظر میگیریم در ابتدا ترد خواندن خط به خط فایل را میخواند و با تشخیص عملگر با اعمال یک wait بر روی سمافور empty عملگر مربوطه در صورت بلاک نشدن (زمانی که بافر عملگر پر نیست) struct که از دو جز عبارت و شماره خط عبارت در فایل ورودی است را در بافر عملگر میگذارد . سپس عملگر با اعمال سمافور مناسب به محاسبه عبارت و سپس ارسال عبارت به بافر نوشتن (در صورت خالی بودن بافر) میکند سپس با اعمال signal به سمافور full نوشتن وارد میشود . و بافر نوشتن نتایج را به همراه شماره خط عبارت در آرایه ای از جنس struct قرار میدهد . در پایان زمانی که همه ترد ها به پایان رسید آرایه را بر اساس شماره خط مرتب و سپس در فایل خروجی به ترتیب می نویسیم .

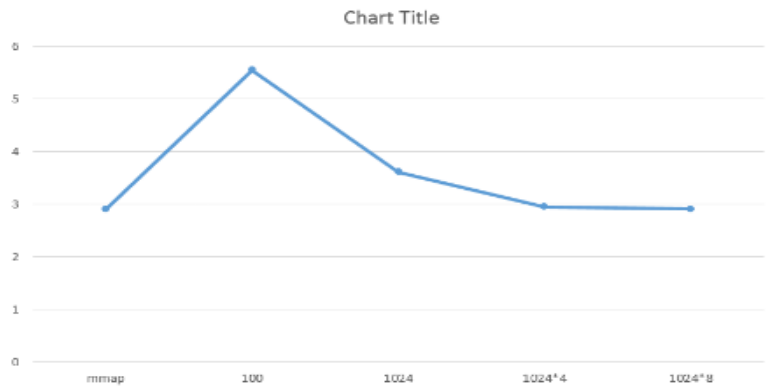
سوال 2) در این سوال به ازای 3 ورودی متفاوت(فایل با حجم کم ، فایل سنگین و فایل باینری سنگین) و با 5 بار اجرای هر کدام از حالت های مختلف chunk و mmap و گرفتن میانگین این نتایج بدست آمده است :

الف)فایل تکست با حجم 97k :



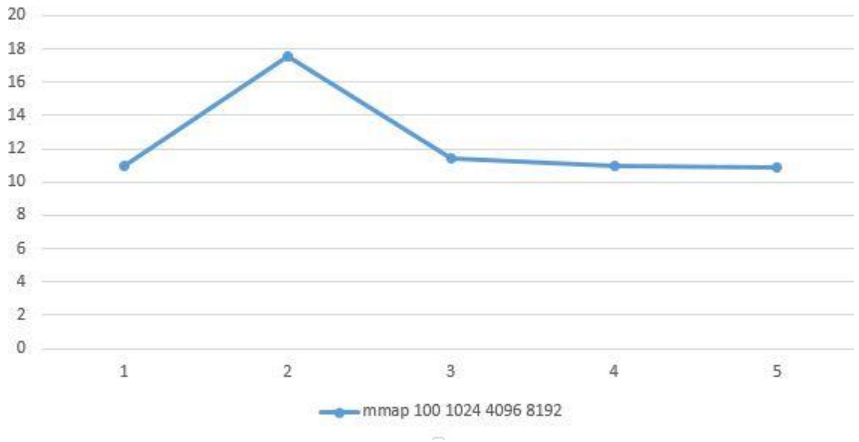
در این نمونه فایل تکست به دلیل حجم کم فایل شاهد ناهنجاری هستیم که دلیل آن میتواند عمل cach کردن به دلیل حجم کم فایل باشد ، بنابرین میتوان نتیجه گیری کرد برای فایل های با حجم کم عمل خواندن به صورت mmap به دلیل اینکه فایل فضای زیادی در حافظه نگرفته و میتواند به طور کامل در حافظه قرار گیرد (به شرط اینکه file size <<<< memory) بهینه ترین حالت میباشد .

ب) فایل تکست با حجم 445MB :



در این نمونه به دلیل حجم زیاد فایل به ازای بزرگتر کردن اندازه chunk زمان اجرا کمتر شده ولی از جایی به بعد (در این جا از 4k) زمان اجرا نسبت به چند برابر شدن اندازه chunk تفاوت بسیار کمتری دارد و حالت شبه نمایی را دنبال میکند. بنابرین میتوان نتیجه گرفت در فایل های نسبتا بزرگ (نسبت به حافظه) روش mmap بهینه ترین حالت نیست و از جایی به بعد که بزرگتر کردن اندازه chunk تاثیر زیادی بر کاهش زمان اجرا ندارد بهینه ترین حالت میباشد (در این نمونه اجرا شده 4k chunk بهینه ترین حالت است زیرا بعد از این حالت افزایش chunk کاهش زمان اجرا کاهش بسیار چشمگیری دارد و بهینه نیست .)

ج) فایل باینری با اندازه 1.3 GB :



در این نمونه در حالی که حجم فایل نسبت به حالت قبل کمتر از 3 برابر شده زمان اجرا تقریبا 6 برابر و حتی بیشتر شده زیرا با افزایش حجم فایل ورودی ممکن است بازوی دیسک برای خواندن بین شیار ها و سکتورهای مختلف جا به شود بنابرین با افزایش حجم فایل افزایش زمان اجرا خطی نخواهد بود و حالت شبه نمایی را دارد. انتخاب حالت بهینه در این نمونه مثل حالت قبل است .