

الگوریتم اراتستن :

این الگوریتم یک الگوریتم نسبتاً خوب برای یافتن اعداد اول از ۲ تا n می باشد .
عملکرد آن به این صورت است که ابتدا اولین عدد اول (۲) را یافته ، سپس از $(۲^۲)$ شروع کرده تمام ضرایب صحیح (۲) را به عنوان غیر اول علامت زده و این روند را تا رادیکال n ادامه میدهد . در آخر الگوریتم ، هر عددی که علامت نخورده باشد (به جز ۰ و ۱) ، عدد اول می باشد .

سیستم عامل لینوکس :

سه مدل کدنویسی تک فرآیندی ، چند فرآیندی و چند ریسمان به پیوست قرار داده شده است که هر سه در سیستم عامل لینوکس پیاده سازی شده است . همچنین با توجه به جدول زیر میتوان دریافت که به طور میانگین برنامه چند فرآیندی دارای بهره وری بهتری از نظر پردازنده دارد .

این در حالی است که برنامه تا فرآیندی از لحاظ بهره وری حافظه ، بهترین شرایط را دارد و آن به این دلیل است که در برنامه های چند فرآیندی یا چند ریسمانی ، ما در واقع حافظه های جدیدی را به خود اختصاص میدهیم . این در حالی است که در برنامه تک فرآیندی این اتفاق نمی افتد .

برنامه چند ریسمانی از لحاظ زمان اجرا ، صاحب بهترین جایگاه است . بهتر بودن آن نسبت به برنامه تک فرآیندی که به وضوح مشخص است . و همچنین چون فرآیند ریسمان سازی ، کاری ساده تر از تولید فرآیند است ، پس در زمان کمتری اجرا شده است .

همه نتیجه های بالا با تعداد زیادی اجرا به دست آمده ولی در شکل زیر ، ۳ مورد آزمایش ها نشان داده شده که حاکی از درست بودن مطالب بالا است .

لازم له ذکر است که کدهای مخصوصی برای بهره وری پردازنده و حافظه در فایل برنامه قرار داده شده که منبع آن ذکر میگردد .

	رتبه اول اجرا	رتبه دوم اجرا	رتبه سوم اجرا
Single Process	time : ۳۹,۲ ms CPU : ۳۳ % Mem : ۱۵۱۰۸	time : ۱۸,۱ ms CPU : ۱۹,۴ % Mem : ۱۴۱۲۸	time : ۳۱,۳ ms CPU : ۳۲,۲ % Mem : ۱۴۹۱۴
Multi Process	time : ۲۱,۹ ms CPU : ۱۱,۵ % Mem : ۱۵۲۲۸	time : ۱۴,۱ ms CPU : ۵,۵ % Mem : ۱۴۱۵۲	time : ۱۷,۹ ms CPU : ۸,۳ % Mem : ۱۴۷۴۰
Multi thread	time : ۲۰,۸ ms CPU : ۲۰ % mem : ۵۰۱۹۲	time : ۹,۳ ms CPU : ۸,۱ % mem : ۴۹۲۱۹	time : ۱۵,۴ ms CPU : ۱۹,۹ mem : ۴۹۷۰۴
	۴۵۵,۵	۲۰۰,۰۰۰	۱۵۰۰,۰۰۰

ویندوز :

در این سیستم عامل هم دقیقا تمام کد ها برنامه نویسی شده است . با توجه به نتایج به دست آمده در شکل زیر ، واضح است که بهره وری در این سیستم عامل نسبت به لینوکس به صورت مشهودی پایین آمده است .

در این جدول نیز مشخص است که برنامه سینگل پراسس از مرتبه زمانی بالاتری نسبت به برنامه های مولتی پراسس و مولتی ترد دارد . همچنین برنامه مولتی پراسس از پردازنده به شدت کمتر استفاده میکند .

میزان استفاده از مموری در برنامه مولتی ترد ، کمتر از بقیه می باشد و این به این دلیل است که از اشتراک مموری استفاده شده همچنین چون در برنامه مولتی ترد مانند مولتی پراسس از کل برنامه کپی نمی شود ، پس استفاده آن از مموری کمتر می باشد .

Subject:	Year:	Month:	Date:	()
	رتبه اول اجرا	رتبه دوم اجرا	رتبه سوم اجرا	
Single Process	time : 22 ms CPU : 25,2 % Mem : 2859008	time : 22,1 ms CPU : 24,2 % Mem : 280576	time : 22 ms CPU : 24,0 % Mem : 2801442	
Multi Process	time : 31 ms CPU : 22,3 % Mem : 272022	time : 32 ms CPU : 22 % Mem : 2809109	time : 32 ms CPU : 22,1 % Mem : 2801442	
Multi Thread	time : 14 ms CPU : 19,1 % Mem : 2015222	time : 12 ms CPU : 18,9 % Mem : 1902792	time : 14 ms CPU : 19,1 % Mem : 192512	