

memory usage :  $\text{multiprocess} > \text{multithread} > \text{singlethread}$

در  $\text{singlethread}$  چون برنامه به خود متوالی و خطی عملی نیست هم اجرای آن شود و هیچ که اضافه ای هم لازم نیست برای زمان اجرای کمتر برنامه نویسم به همین دلیل memory کمتری مصرف می کند

در  $\text{multithread}$  با برنامه راضی که می کنیم و هر کدام به خود متوالی با دیگری اجرای آن شود و resource code مشترک دارد که در  $\text{multiprocess}$  این طوری نیست

در  $\text{multiprocess}$  ، در هر یک یک سیستم (  $\text{create process}$  یا  $\text{fork}$  )

که فایده دوباره گیری می شود و با نوشتن تابع (  $\text{exe}$  ) که های اضافی (عده بریده) را می نویسم به همین دلیل  $\text{multiprocess}$  فضای حافظی خیلی بیشتری نسبت به  $\text{multithread}$  هر فرد

این نسخه هم در  $\text{linux}$  هم در  $\text{windows}$  برقرار است

cpu utilization :  $\text{multithread} > \text{multiprocess} > \text{singlethread}$

هرچه میزان بهره داری از CPU بیشتر باشد مطلوب تر است

در  $\text{multithread}$  چون  $\text{swapping}$  نداریم و source code ها مشترک هستند ،

به همین دلیل بهره داری از سیستم بالا تر است نسبت به  $\text{multiprocess}$  که هم  $\text{swapping}$  دارد ، هم آنکه source code ها مشترک نیستند

یعنی بهره داری این ها از  $\text{singlethread}$  بهره داری بیشتری دارند ، زیرا  $\text{singlethread}$

تو یک زمان فقط یک برنامه دارد اجرای آن شود و نتیجه میزان استفاده آن از CPU کمتر است

زبان: به طور کلی در linux زبان اجرایی برنامه ها خنجر کمتر از windows است  
۳- دلیل: تخصیص حافظه ها می متفاوت (malloc)  
۲- exception handling

۳- OS به مقداردهی اولیه وابسته است. به عبارتی این دو ویژگی در کنار هم باید  
هنگام اجرای برنامه انجام شود و در صورتی که انجام نشود، سرعت و پایداری برنامه  
تغایر میکند. زبان multi thread > single thread > multiprocess  
تغییر می کند. زبان اجرایی single به پایداری multi thread, multiprocess  
کمتر وابسته. زیرا multi thread به صورت موازی چند تکه برنامه را اجرا می کند و  
multiprocess به صورت همزمان برنامه را اجرا می کند.

- مقایسه زبان multi thread, multiprocess: زیرا کامپیوترهای امروزی و سیستم  
(لب تاب - موبایل) دارای ۸ core یا کمتر است. به همین دلیل process های متعدد روی  
core ها به هم می افتد و در multi thread روی CPU اجرا می شود.  
به همین دلیل زبان multiprocess کمتر از multi thread است.