MULTIPROCESS CONCURRENT QUICK SORT

[

->In normal quick sort we divide array into two parts and than call function again on left part and than right part.

->In multiprocess quick sort we divide array into two part than fork the process to apply sort on left and right part.

->We store the array in shared mermory so it can be accesed by all the child process.

->The parent process waits until both the child process for left and right are complted by using waitpid();

]

MULTITHREADED QUICK SORT

[

-> In multithreaded quick sort we divide array into two parts than create new threads to apply sort on left and right part.

-> The parent thread joins the two child thread unsing thread_join()

]

Than we compared the various methods by comparing the time taken in sorting.

Ex input:-

```
5
7 2 9 1 4

Running concurrent_quicksort
time = 0.000004

Running threaded_concurrent_quicksort
time = 0.000263

Running normal_quicksort
time = 0.000003
normal_quicksort ran:
```

```
[ 1.260873 ] times faster than concurrent_quicksort
[ 81.604300 ] times faster than threaded_concurrent_quicksort
```