بسمه تعالی

سیستم های بی‌درنگ
گزارش فاز اول پروژه

پوریا عارفی جمال
علیرضا فرودنیا

شرح الگوریتم ها:

الگوریتم Co operative:

این الگوریتم همواره یکی از تسک ها که در آرایه Tasks جلوتر از بقیه قرار دارد را انتخاب کرده و تمام هسته هایی که در اختیار دارد را بدان اختصاص می‌دهد. از آنجایی که سیستم کلا دارای ۶ هسته می‌باشد این الگوریتم معادل اجرای FCFS بر روی ۶ هسته به صورت موازی می‌باشد.

الگوریتم Best:

این الگوریتم در واقع کل فضای حالات مساله را بررسی می‌کند و در آخر بهترین زمان بندی را ارائه می‌دهد.
شیوه اجرای این الگوریتم به این صورت است که ابتدا T عدد Task ورودی را گرفته و تمام حالات مختلف اجرای آنها (اجرای هر کدام روی تعدادی core) را در نظر می‌گیریم که می شود ۶ به توان T. سپس به ازای هر کدام از این زیر مساله ها !T بار یک الگوریتم greedy را اجرا می‌کنیم که به صورت FCFS تسک ها را اجرا کند. در آخر بهترین زمان اجرای تسک ها حاصل می‌شود.

الگوریتم Profile:

در این الگوریتم در ابتدای مساله یک آرایه profiles که همان میزان S(I,J)I ها می‌باشد محاسبه می‌شود. سپس در ادامه یک الگوریتم مشابه با knapsack اجرا می‌شود که اگر تعداد تسک ها از core ها بیشتر باشد، ارزشمند ترین profile های ممکن را انتخاب کند که همواره برابر S(I,1)I می‌باشد، و هر زمان که تعداد تسک ها از core

ها کمتر شود آن تسک که در صورت اجرا روی چند core بیشترین speed up را دارد بر روی چند core اجرا می‌شود و سایر تسک ها بر روی یک core اجرا می‌شوند.

شکل زمان بندی تمام الگوریتم های بالا با اجرای کد مشخص است.

ساختار کد:

برای اجرای الگوریتم های بالا یک class با نام TaskScheduler در کد وجود دارد که در ابتدا با صدا زده شدن تابع generate_tasks به تعداد داده شده به آن، از میان تسک های موجود در CSV ها انتخاب می کند.

سپس توابع مرتبط با هرکدام از الگوریتم ها با نام خودشان قرار دارند که در صورت صدا زده شدن، الگوریتم مذکور را اجرا کرده و نمودار مربوط به آن را ترسیم می‌کنند.

نتایج هرکدام از الگوریتم ها در یک class با نام Result ذخیره می‌شود تا بعدا در دسترس باشد.

همچنین یک class کمکی با نام Core وجود دارد که اجرا شدن یک Task بر روی یک هسته از یک زمان شروع تا پایان را در خودش نگاه می‌دارد تا در کشیدن نمودار زمان‌بندی استفاده شود.