



به نام خدا

پروژه ی شبیه سازی

استاد :

دکتر صفایی

اعضای گروه :

علی احمدی کافشانی

۹۷۱۰۵۷۰۳

علی جوانمرد

۹۷۱۰۵۸۷۷

دانشگاه صنعتی شریف

زمستان ۱۴۰۱

گزارش پروژه

در این پروژه بر آن بودیم که یک cpu تک هسته را با دو لایه صف شبیه سازی نماییم. از کتابخانه `simpy` در پایتون بهره بردیم و برای پیاده سازی منطق و تولید نمونه تصادفی از `numpy` استفاده کردیم.

پروژه از چند بخش تشکیل شده است که به ترتیب به این موارد می پردازیم:

۱. در بخش `run_simulation` ابتدا یک محیط `simpy` ساخته می شود سپس به ترتیب کار هایی که برای شبیه سازی نیاز داریم انجام میشوند. ابتدا صف لایه ی اول ، سپس صف لایه ی دوم بعد تسک ها و تابع انتقال تسک به لایه دوم اجرا می شوند و در انتها خروجی شبیه سازی چاپ میگردد.

۲. یک `process` به نام `layer2queue` داریم که هسته اصلی ما را تشکیل میدهد. این پردازش از ۳ صف تشکیل شده است. ابتدا تسک ها به صف `RR` اول میروند و سپس به صف های بعد بر حسب نیاز منتقل می شوند. در هر مرحله از شبیه سازی در این `process` ابتدا با احتمال های داده شده یکی از این سه صف را انتخاب میکنیم و سپس اولین تسک در این صف (تسکی که مدت بیشتری در صف بوده را انتخاب کرده و به `cpu` می دهیم تا آن را اجرا کند.

این اجرا با استفاده از تابع `timeout` در کتابخانه `simpy` شبیه سازی میشود. همچنین قبل از اجرا بررسی میشود که زمان باقی مانده تسک از حداکثر مجاز بیشتر نشود و تسک اگر `expire` شده است یا بعد از انجام در این مرحله `expire` میشود اجرا نشود. همچنین در آخر بعد از اجرا این تسک به تسک های تمام شده اضافه میشود تا گزارش گیری انجام شود.

۳. در بخشش `job loader` وظیفه این را دارد که در صورت وجود تسک در لایه اول از صف لایه ی اول `k` تسک با بیشترین اولویت را به لایه دوم منتقل کند.

۴. در بخش job creator به فاصله زمانی توزیع پواسن با میانگین X تسک ها را می سازد و وارد لایه ی اول میکند.

۵. در بخش Result creator مقدار های خروجی تست را نمایش می دهد که شامل طول متوسط صف و مقدار میانگین انتظار در صف ۱ و ۲ ، مقدار بهره‌وری cpu و در آخر تعداد تسک های منقضی شده را نشان می دهد. این خروجی ها هر ۱ واحد زمانی بررسی و جمع آوری میشوند تا در انتهای شبیه سازی گزارش گیری انجام گردد.

۶. در انتها یک بخش بهینه سازی روی پارامترهای مختلف برای T1 و T2 بر روی تسک های برابر انجام میشود تا بهترین گزینه را پیدا کنیم.

اجرای تست شبیه سازی

Run

Avarage Queue Length -> 0.82

Avarage Time in line 1 -> 7.3

Avarage Time in line 2 -> 10.1

Cpu utilization -> 0.472

Expired Tasks -> 80.0 %

----- Run 1 -----

Avarage Queue Length -> 0.494

Avarage Time in line 1 -> 5.375

Avarage Time in line 2 -> 6.625

Cpu utilization -> 0.472

Expired Tasks -> 87.5 %

----- Run 2 -----

Avarage Queue Length -> 1.348

Avarage Time in line 1 -> 12.0

Avarage Time in line 2 -> 13.7

Cpu utilization -> 0.573

Expired Tasks -> 100.0 %

----- Run 3 -----

Avarage Queue Length -> 0.124

Avarage Time in line 1 -> 1.1

Avarage Time in line 2 -> 4.9

Cpu utilization -> 0.27

Expired Tasks -> 20.0 %

----- Run 4 -----

Avarage Queue Length -> 1.303

Avarage Time in line 1 -> 11.6

Avarage Time in line 2 -> 15.1

Cpu utilization -> 0.697

Expired Tasks -> 90.0 %

----- Optimization Run with T1=1 T2=8 -----

Avarage Queue Length -> 0.427

Avarage Time in line 1 -> 3.8

Avarage Time in line 2 -> 5.9

Cpu utilization -> 0.258

Expired Tasks -> 80.0 %

----- Optimization Run with T1=1 T2=10 -----

Avarage Queue Length -> 0.472

Avarage Time in line 1 -> 4.2

Avarage Time in line 2 -> 6.5

Cpu utilization -> 0.281

Expired Tasks -> 80.0 %

----- Optimization Run with T1=1 T2=12 -----

Avarage Queue Length -> 0.517

Avarage Time in line 1 -> 4.6

Avarage Time in line 2 -> 7.1

Cpu utilization -> 0.303

Expired Tasks -> 80.0 %

----- Optimization Run with T1=2 T2=8 -----

Avarage Queue Length -> 0.652

Avarage Time in line 1 -> 5.8

Avarage Time in line 2 -> 7.8

Cpu utilization -> 0.494

Expired Tasks -> 70.0 %

----- Optimization Run with T1=2 T2=10 -----

Avarage Queue Length -> 0.652

Avarage Time in line 1 -> 5.8

Avarage Time in line 2 -> 7.8

Cpu utilization -> 0.517

Expired Tasks -> 70.0 %

----- Optimization Run with T1=2 T2=12 -----

Avarage Queue Length -> 0.652

Avarage Time in line 1 -> 5.8

Avarage Time in line 2 -> 6.6

Cpu utilization -> 0.382

Expired Tasks -> 60.0 %

----- Optimization Run with T1=3 T2=8 -----

Avarage Queue Length -> 0.393

Avarage Time in line 1 -> 3.5

Avarage Time in line 2 -> 6.4

Cpu utilization -> 0.371

Expired Tasks -> 50.0 %

----- Optimization Run with T1=3 T2=10 -----

Avarage Queue Length -> 0.393

Avarage Time in line 1 -> 3.5

Avarage Time in line 2 -> 6.4

Cpu utilization -> 0.393

Expired Tasks -> 50.0 %

----- Optimization Run with T1=3 T2=12 -----

Avarage Queue Length -> 0.393

Avarage Time in line 1 -> 3.5

Avarage Time in line 2 -> 6.4

Cpu utilization -> 0.393

Expired Tasks -> 50.0 %

----- Optimization Run with T1=5 T2=8 -----

Avarage Queue Length -> 0.36

Avarage Time in line 1 -> 3.2

Avarage Time in line 2 -> 4.8

Cpu utilization -> 0.393

Expired Tasks -> 60.0 %

----- Optimization Run with T1=5 T2=10 -----

Avarage Queue Length -> 0.36

Avarage Time in line 1 -> 3.2

Avarage Time in line 2 -> 4.8

Cpu utilization -> 0.393

Expired Tasks -> 60.0 %

پیوست

آدرس گیت پروژه: https://github.com/aliraad79/Cpu_Simulater