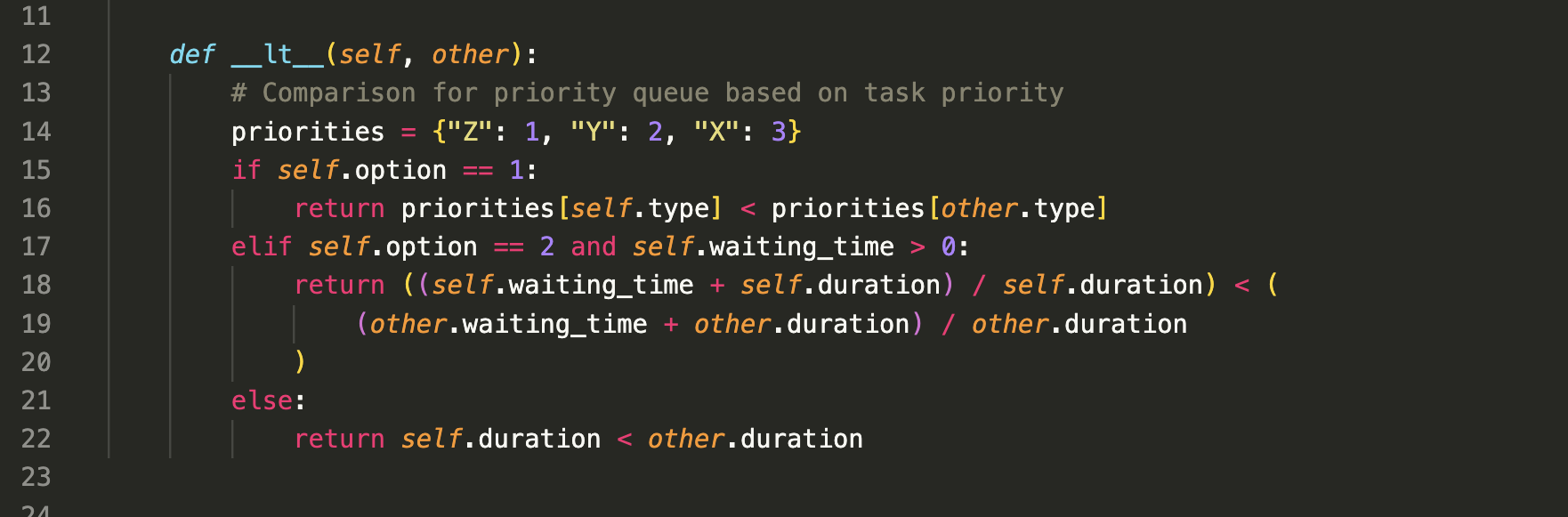
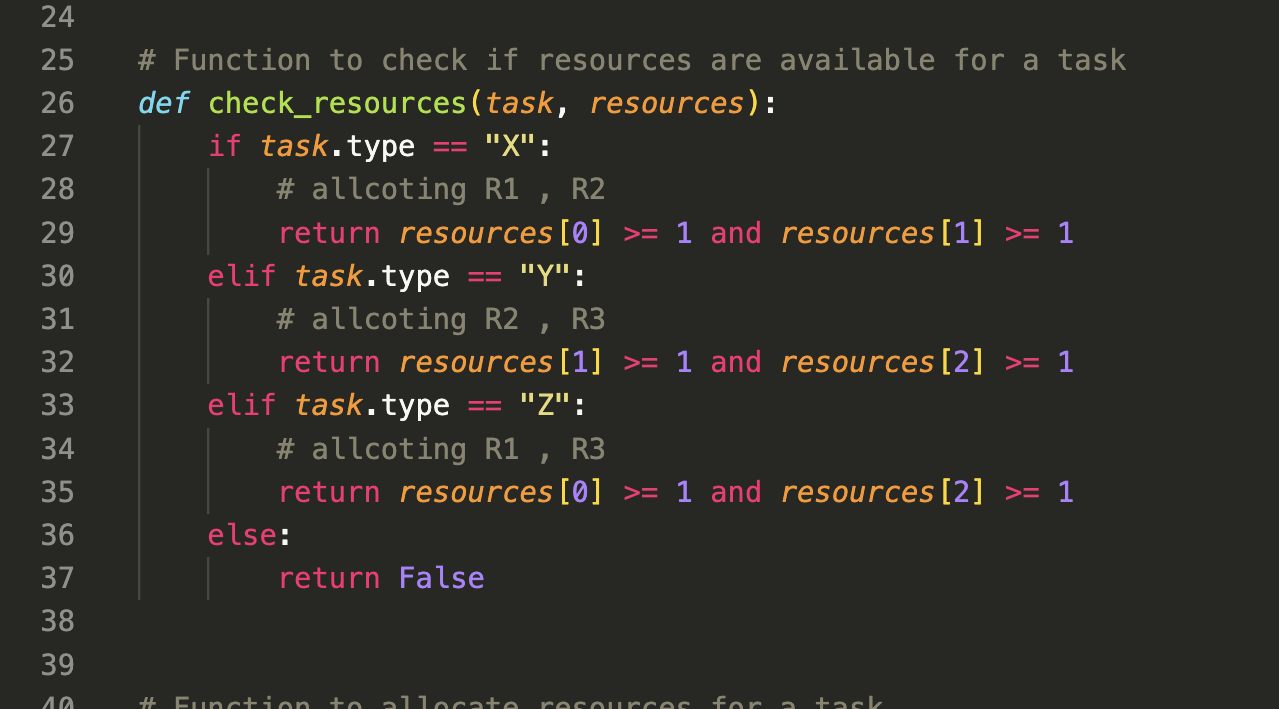
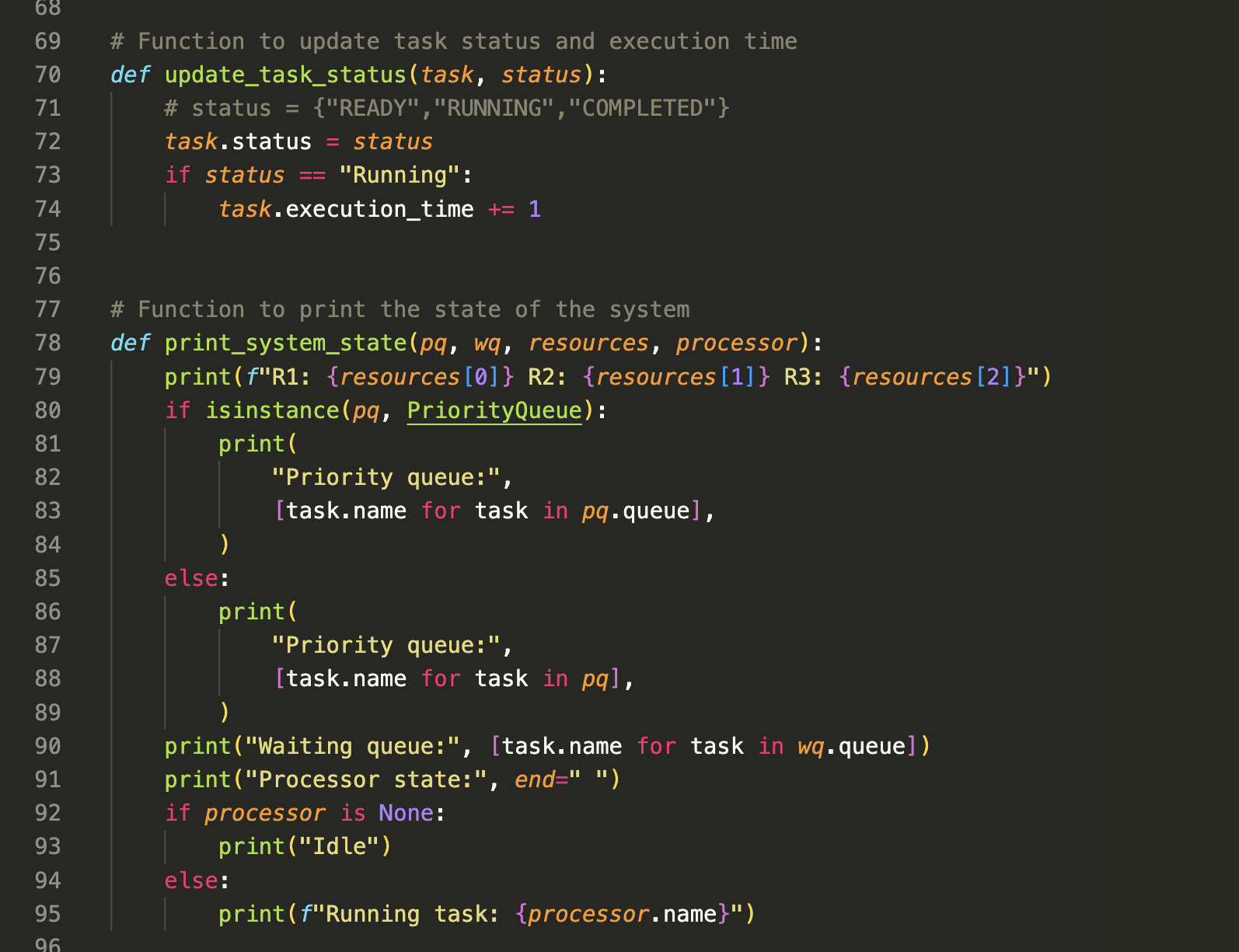
پروژه سیستم

در ابتدا **تابع iteration** در کلاس task در واقع برای بررسی اولویت تسک های ورودی براساس نوع زمانبندی انتخاب شده توسط کاربر هست که option 1 برای RR کاربرد دارد همانطور که تعریف شده، option 2 برای چیندن تسک ها به ترتیب اولویت ضریب الگوریتم HRRN که همانطور که در کد نشان داده شده است محاسبه شده و کمترین مقدار دارای اولویت بالاتر می باشد.

در غیر این صورت برای SJF که بخواهیم سورت کنیم، بر اساس زمان burst که کمترین زمان اجرا را اولویت بالاتر قرار میدهد.

**تابع check\_resources** برای برسی اینکه برای هر نوع تسک آیا از ۳ تا منبع R1,R2,R3 منابع مورد نیاز خودش دردسترسش هست که در این صورت true برمیگرداند در غیر اینصورت اعلام میکند که منابع مورد نیاز در دسترس نمی باشد و false میشود.

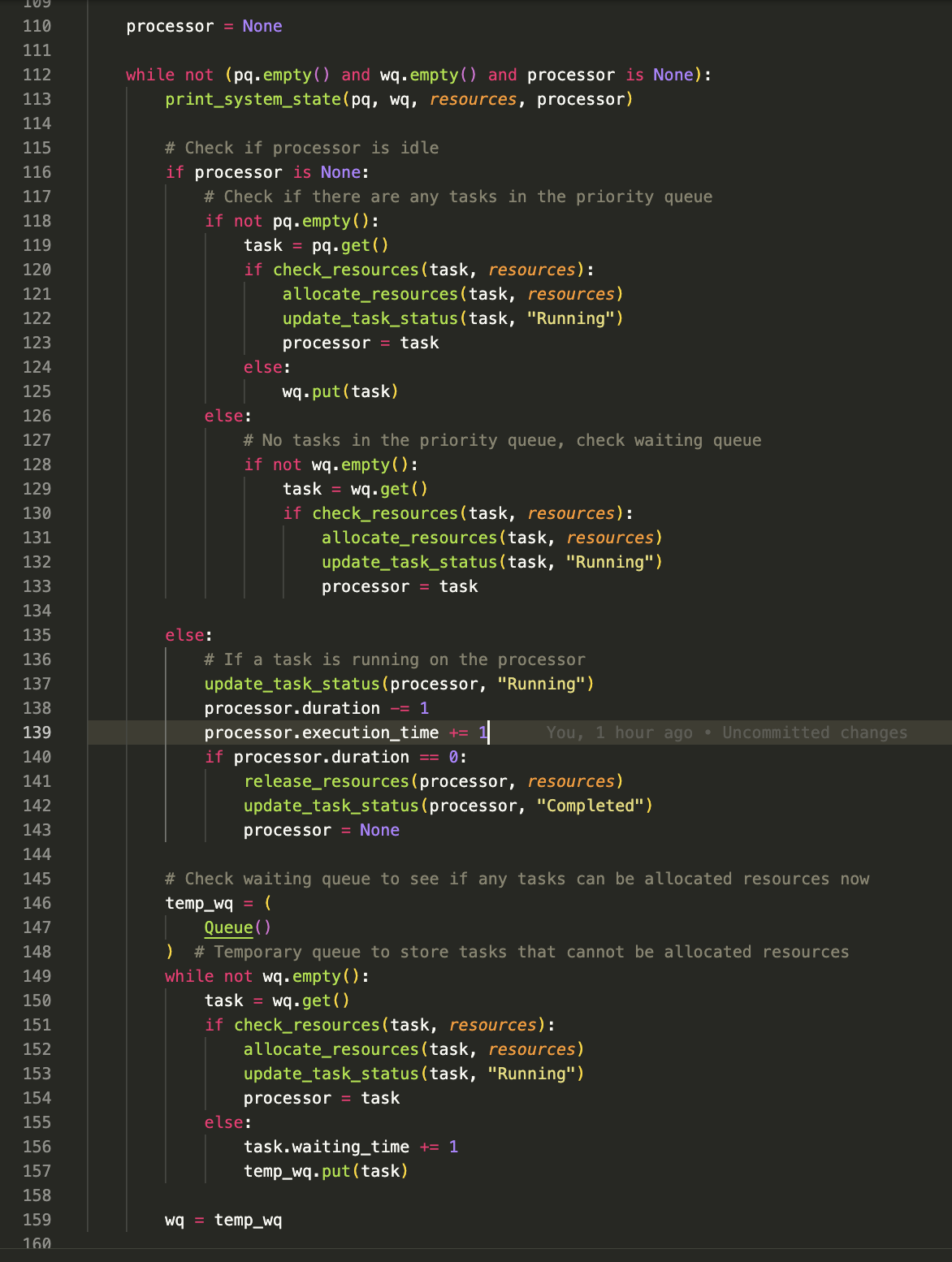
**توابع allocate\_resources و release\_resources** برای تخصیص منابع مورد نیاز و آزاد کردن منابع در اختیار برای هر تسک تعریف شده است.



**تابع update\_task\_status** برای بررسی حالت تسک می باشد که زمان اجرای برنامه را بروزرسانی میکند.

برای نمایش صف ها،عبارت isinstance(pq, PriorityQueue) برای بررسی این استفاده می‌شود که آیا متغیر pq یک نمونه از کلاس PriorityQueue است یا خیر چون برای بعضی زمانبندی ها از صف اولویت برای اولویت بندی تسک ها استفاده شده است.

زمانبندی ها



در تمامی زمانبندی ها برای اجرای برنامه تا زمانی انجام میشود که یا فرایند در پردازنده نباشد و یا صف ها خالی شده بدر برنامه‌ای که ارائه داده شده است، pq و wq به ترتیب به یک PriorityQueue و یک Queue مقداردهی می‌شوند.

در SJFوHRRN،RR که نیاز به اولویت بندی بین تسک ها می باشد، pqبه عنوان یک PriorityQueue استفاده می‌شود و در آن تمام وظایف (tasks) براساس مدت زمان(SJF) و ضریب اولویت (HRRN)آن‌ها مرتب می‌شوند. وظایف به ترتیب اولویت تعیین شده در iteration آن‌ها در صف قرار می‌گیرند. سپس با استفاده از تابع put، هر وظیفه به pq اضافه می‌شود.

wq به عنوان یک Queue استفاده می‌شود و وظایفی که نمی‌توانند در حال حاضر منابع را دریافت کنند، در آن قرار می‌گیرند. وظایف در صف wq با استفاده از تابع put قرار می‌گیرند.

به طور خلاصه، pq برای مرتب سازی و انتخاب وظایف براساس مدت زمان، ضریب ratio و اولویت نوع تسک آن‌ها به کار می‌رود، و wq برای نگهداری وظایفی که در حال حاضر نمی‌توانند منابع را دریافت کنند، استفاده می‌شود.

توضیحات هر حلقه و شرط در کامنت ها ذکر شده اند.

SJF

.1تابع schedule\_sjf وظایف را با استفاده از الگوریتم Shortest-Job-First (SJF) زمان بندی می کند.

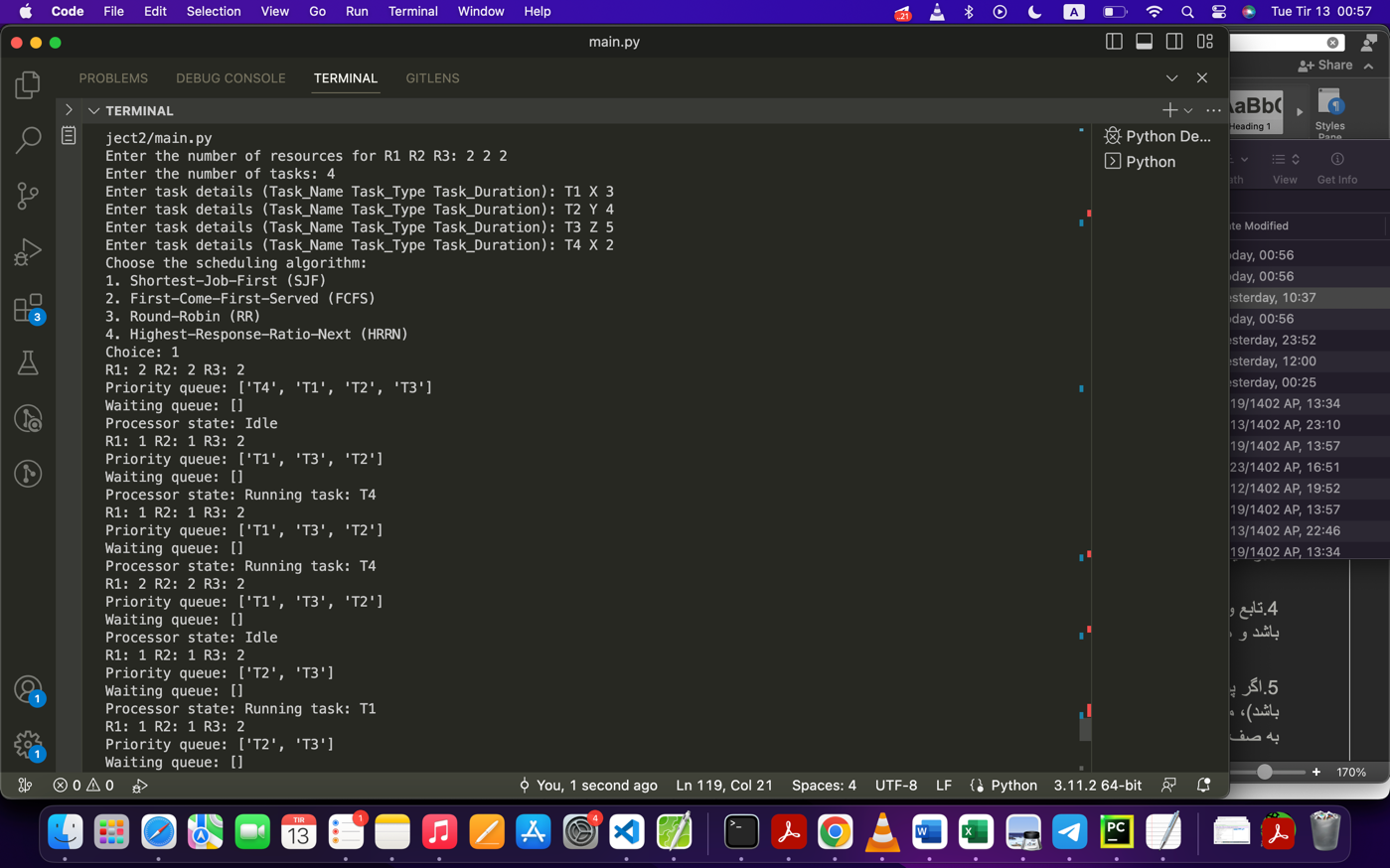
.2با مرتب کردن وظایف بر اساس مدت آنها به ترتیب صعودی شروع می شود.

.3وظایف بر اساس مدت آنها به صف اولویت (pq) اضافه می شوند.

.4تابع وارد یک حلقه می شود که تا زمانی ادامه می یابد که هیچ وظیفه ای در صف اولویت (pq)، صف انتظار (wq) وجود نداشته باشد و هیچ وظیفه ای در حال حاضر روی پردازنده اجرا نشود.

.5اگر پردازنده بیکار باشد، صف اولویت (pq) را برای وظایف بررسی می کند. اگر وظیفه مناسبی پیدا شود (یعنی منابع آن در دسترس باشد)، منابع را تخصیص می دهد، وضعیت کار را به روز می کند و آن را به پردازنده اختصاص می دهد. در غیر این صورت، وظیفه به صف انتظار (wq) اضافه می شود.

.6 اگر پردازنده قبلاً وظیفه ای را اجرا می کند، آن کار را با کاهش مدت زمان آن و به روز رسانی زمان اجرای آن اجرا می کند. اگر مدت زمان کار به صفر برسد، به عنوان تکمیل شده علامت گذاری می شود، منابع آزاد می شوند و پردازنده بیکار می شود.

.7پس از رسیدگی به وظیفه در حال اجرا، کد صف انتظار (wq) را بررسی می کند تا ببیند آیا می توان به وظایفی منابع اختصاص داد یا خیر. اگر منابع در دسترس باشد، یک وظیفه انتخاب می شود، منابع تخصیص داده می شود و به پردازنده اختصاص می یابد. وظایفی که نمی توانند منابع را تخصیص دهند در یک صف موقت نگه داشته می شوند.

.8سپس صف انتظار با وظایف از صف موقت به روز می شود و حلقه ادامه می یابد.

.9پس از اتمام حلقه، تمام وظایف برنامه ریزی شده است، و الگوریتم SJF کامل شده است.

بطور مثال طبق ورودی داده شده، ترتیب صف اولویت تشکیل شده به شکل مقابل می باشد.

FCFS

1. صف اولویت (pq) و صف انتظار (wq) ایجاد می‌شوند.

2. وظیفه‌ها به ترتیب ورود به صف اولویت (pq) افزوده می‌شوند.

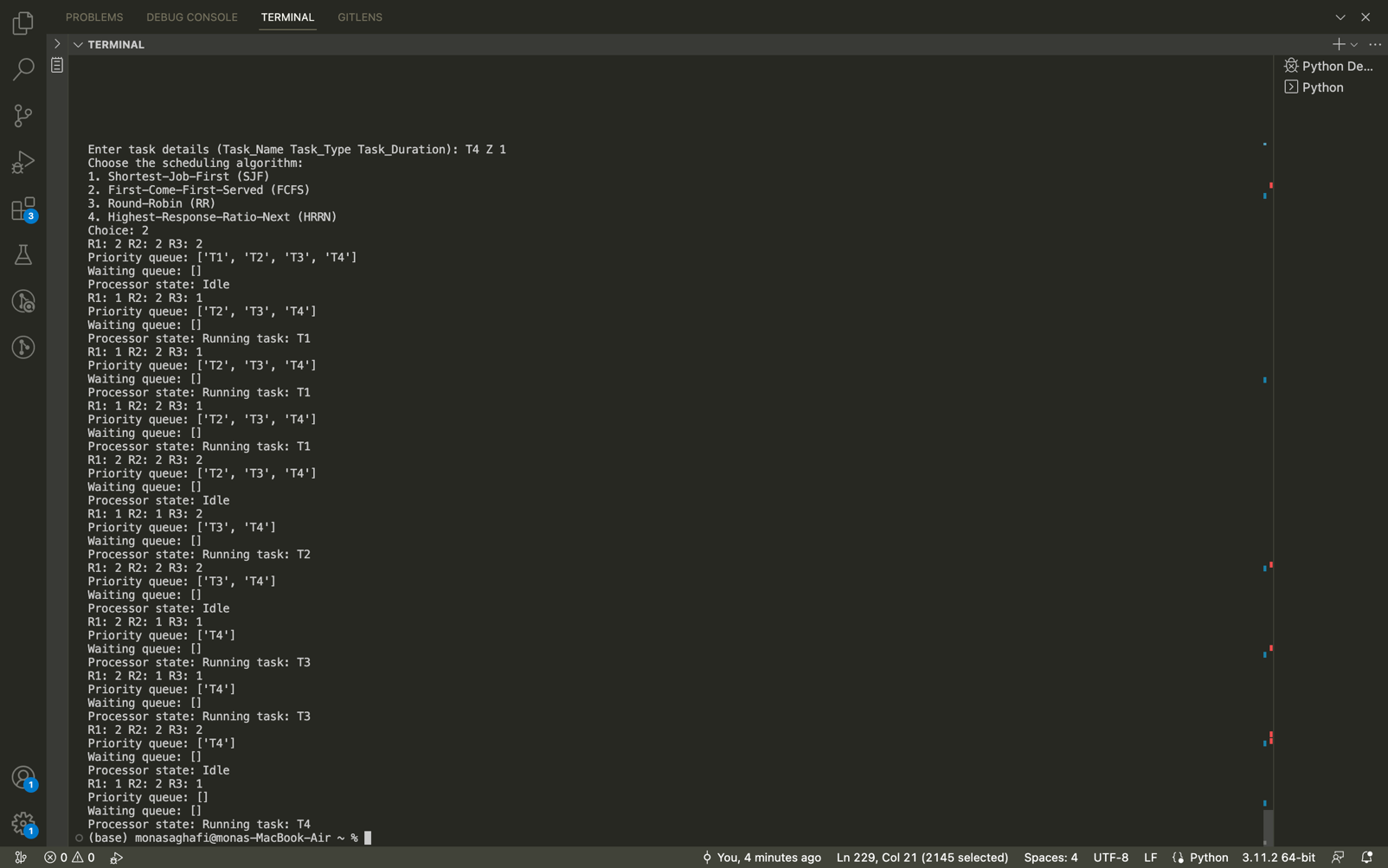
3. یک پردازشگر (متغیر processor) مقداردهی اولیه می‌شود.

4. یک حلقه ایجاد می‌شود که تا زمانی که صف اولویت (pq) و صف انتظار (wq) خالی بوده و پردازشگر در حالت آزاد نبوده، ادامه می‌یابد.

5. در هر دور از حلقه، وضعیت فعلی سیستم چاپ می‌شود و میزان استفاده پردازشگر نیز نمایش داده می‌شود.

6. اگر صف اولویت (pq) وظیفه‌ای دارد و پردازشگر در حالت آزاد است، وظیفه اولیه از ابتدای صف حذف می‌شود و به پردازشگر اختصاص داده می‌شود.

7. اگر پردازشگر در حالت آزاد نباشد، کد وظیفه‌های در صف اولویت (pq) و صف انتظار (wq) را بررسی می‌کند و آن‌ها را به ترتیب چک می‌کند.

8. اگر وظیفه ای در صف اولویت (pq) وجود داشته باشد و پردازشگر در حالت آزاد نباشد:

- اگر منابع لازم برای اجرای وظیفه در دسترس باشند، وظیفه از صف حذف می‌شود و به پردازشگر اختصاص داده می‌شود.

همانطور که میبینید تسک های داده شده بترتیب اجرا شده اند.

RR

در زمانبندی Round Robin (RR) گرسنگی در الگوریتم RR با مفهوم کوانتوم زمانی کنترل می شود. در RR به هر وظیفه یک کوانتوم زمان ثابت برای اجرا روی پردازنده داده می شود. اگر یک کار در کوانتوم زمانی خود کامل نشود، به طور موقت از پردازنده حذف می شود و در انتهای صف (چه صف اولویت یا صف انتظار) قرار می گیرد تا به سایر وظایف اجازه اجرا داده شود. این امر مانع از انحصار هر کار واحد در پردازنده برای مدت طولانی می شود.

این کد همچنین با پیاده سازی مفهوم صف انتظار (wq) تا حدی از گرسنگی وظایف مدیریت می کند. اگر به یک کار به دلیل محدودیت منابع نتوان فوراً منابع را تخصیص داد، در صف انتظار قرار می گیرد و به آن اجازه می دهد تا زمانی که منابع مورد نیاز در دسترس قرار گیرد منتظر بماند. این تضمین می کند که وظایف به دلیل در دسترس نبودن منابع به طور نامحدود گرسنه نمی شوند.

به طور کلی، الگوریتم RR و مکانیسم صف انتظار مورد استفاده در کد ارائه شده با ارائه زمان اجرای منصفانه برای هر کار و اولویت بندی وظایف در صف بر اساس زمان رسیدن و در دسترس بودن منابع، به کاهش گرسنگی کمک می کند.

همچنین برای زمان t تعریف شده در کد، تا زمانی که یکی از تسک ها بطور کامل روی پردازنده اجرا شود، افزایش میابد و برای جلوگیری از تداخل محاسبه زمانی با تایم کوانتوم، مجدد آن ۰ میشود.

نمونه ی خروجی آن در فایل txt قابل مشاهده می باشد.

HRRN

در الگوریتم، وظیفه‌ها بر اساس نسبت پاسخ دهی بالاتر انتخاب می‌شوند. دقیق‌تر، اولین وظیفه با بیشترین نسبت پاسخ دهی در صف اولویت (pq) انتخاب می‌شود و به طور خاص در صف حاوی اولین وظیفه با نسبت بالاتر قرار دارد. هر زمان که وظیفه‌ای از پردازشگر خارج می‌شود، نسبت پاسخ دهی وظیفه‌ها به روز رسانی می‌شود و ترتیب صف اولویت نیز تغییر می‌کند.

در هر دور از حلقه، حالت سیستم چاپ می‌شود و سپس بررسی می‌شود که آیا پردازشگر خالی است یا خیر. اگر پردازشگر خالی باشد، اولین وظیفه با نسبت پاسخ دهی بالاتر از صف اولویت (pq) انتخاب می‌شود و منابع لازم به آن تخصیص داده می‌شود. اگر صف اولویت خالی باشد، وظیفه‌های در حال انتظار در صف انتظار (wq) بررسی می‌شوند و وظیفه با نسبت بالاتر انتخاب شده و منابع به آن تخصیص داده می‌شود. گر پردازشگر در حالت اجرا باشد، محتویات پردازشگر بروزرسانی شده و زمان اجرای آن به اندازه یک واحد کاهش می‌یابد. سپس بررسی می‌شود که آیا زمان اجرای وظیفه به پایان رسیده است یا خیر. اگر زمان اجرای وظیفه به پایان رسیده باشد، وظیفه مورد نظر از پردازشگر خارج شده و منابع آن آزاد می‌شوند. وضعیت وظیفه به "تمام شده" تغییر می‌کند و پردازشگر را خالی می‌کند سپس در حلقه، به همه‌ی وظیفه‌ها پیمایش می‌شود و در صورتی که وظیفه در حال حاضر وظیفه در حال اجرا نباشد، زمان انتظار آن وظیفه افزایش می‌یابد.