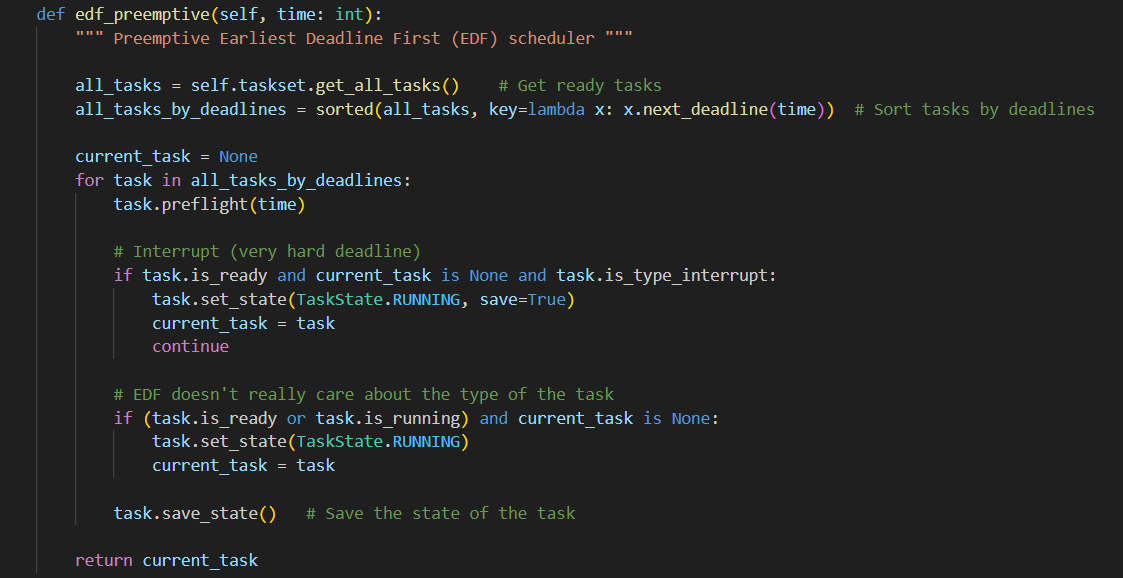
# پیاده سازی

پیاده سازی scheduler و سیستم RTOS از ابتدا انجام شده (:

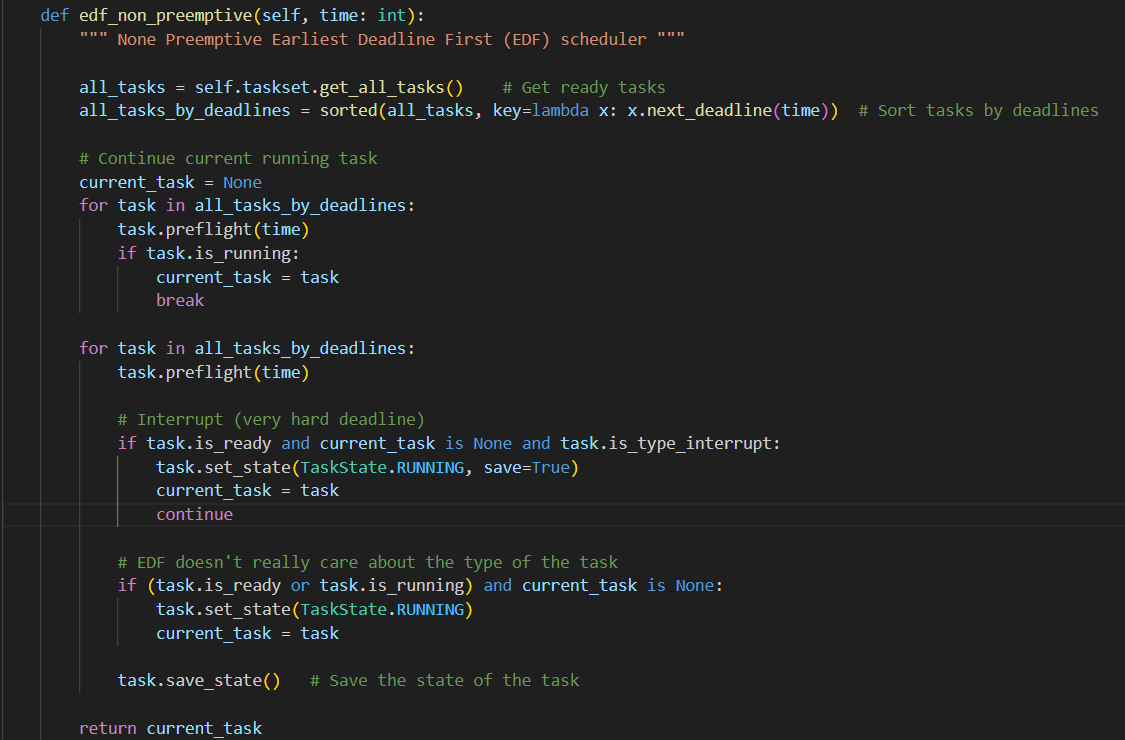
## پیاده سازی Preemptive EDF

برای این الگوریتم تنها نزدیک ترین زمان deadline به صورت داینامیک اهمیت دارد و نوع تسک اهمیت چندانی ندارد (جز تسک از نوع interrupt که اولویت همیشه بالاترین را دارد و برای آن در کد استثنا قائل شدیم).



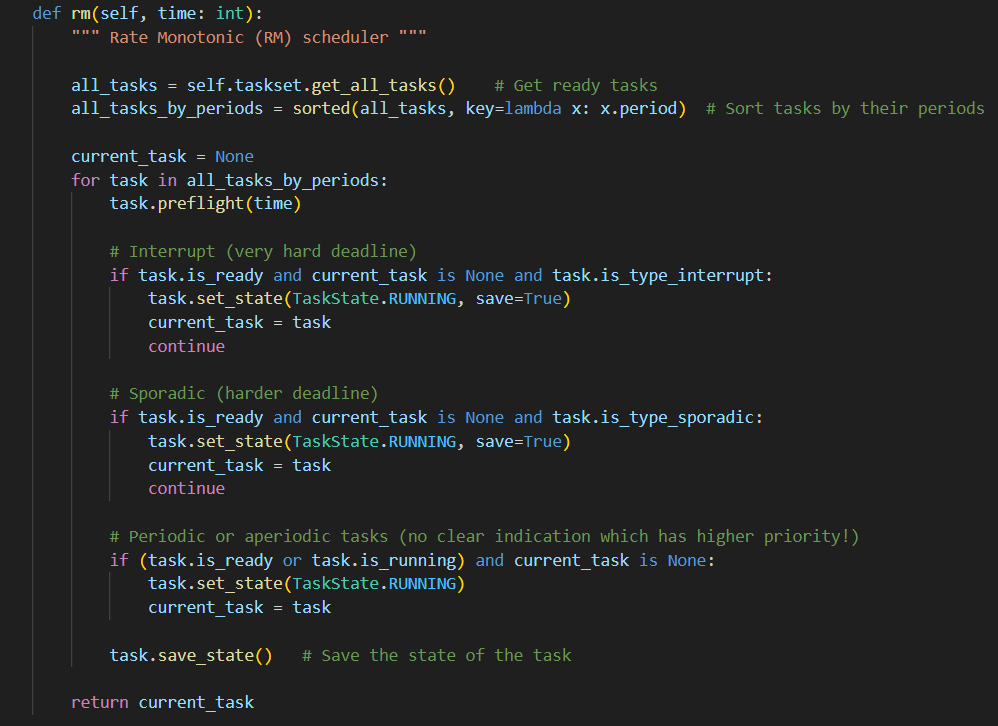
## پیاده سازی Non-Preemptive EDF

این الگوریتم نیز مشابه الگوریتم بالا بوده با این تفاوت که تسک در حالت RUNNING، باید RUNNING باقی بماند تا اجرا آن تکمیل شود.



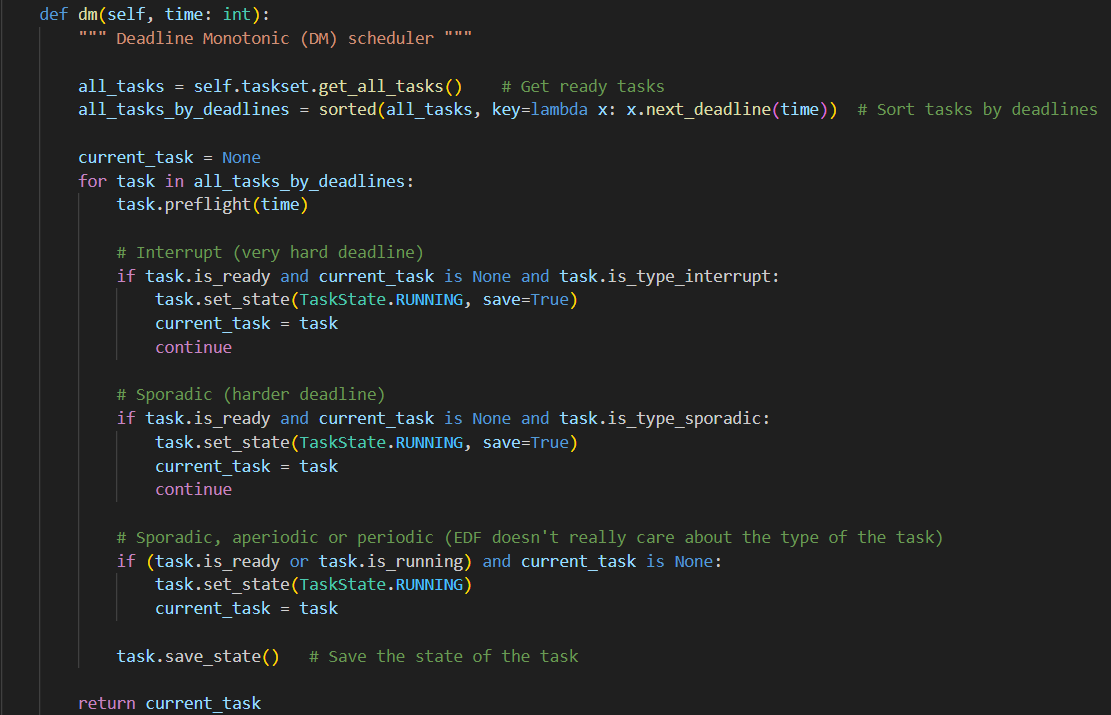
## پیاده سازی RM

این الگوریتم static بوده و برای تسک های hard deadline (جز interrupt که همیشه بالاترین اولویت است) اهمیت بالاتری قائل است. این الگوریتم بر اساس کوتاه ترین Period کار می‌کند و از آن جایی که static بوده، بر خلاف EDF در حال اجرا زمان period ارزیابی نمیشود.



## پیاده سازی DM

این الگوریتم هم مانند RM اما بر روی نزدیک ترین ددلاین به صورت static کار می‌کند. الگوریتم ها تقریبا شبیه به هم هستند جز در پارامتر های تصمیم گیری آنها.

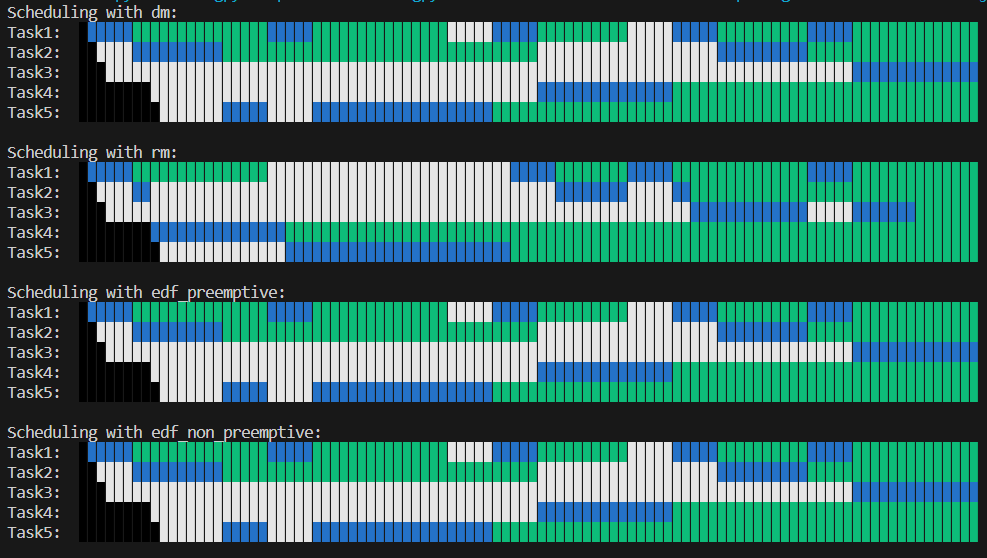


# در هر تسک ست داده شده کدام الگوریتم مناسب تر است؟

بر حسب شبیه سازی های انجام شده توسط کد خروجی زیر حاصل شد:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P-EDF | NP-EDF | DM | RM | Task Set / Optimality |
| Yes | Yes | Yes | Yes | Task1 |
| Yes | Yes | Yes | Yes | Task2 |
| Yes | Yes | Yes | Yes | Task interrupt |

## خروجی task1



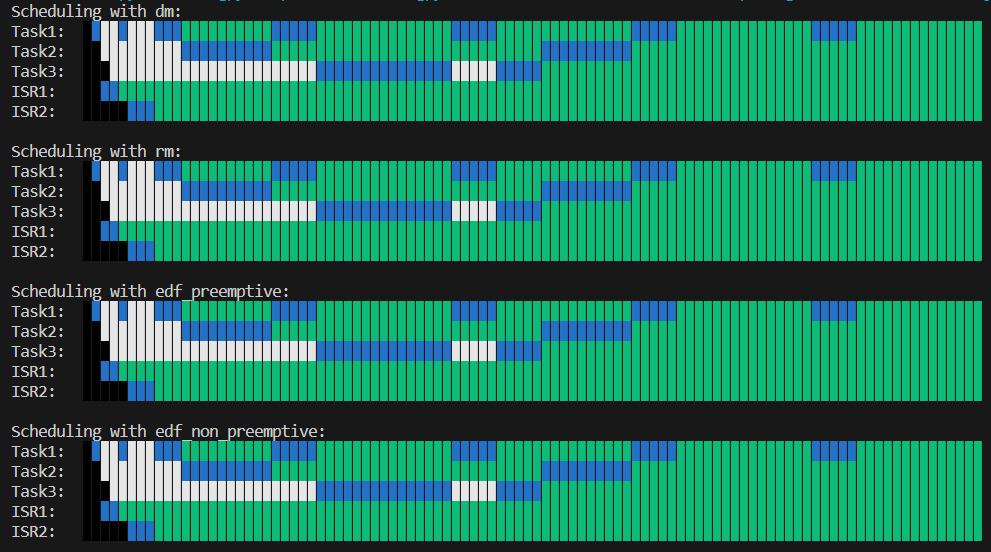
همانطور که با SimSo تایید شد (خروجی یکسان است)، RM برای تسک اول فیل شده و DM و EDF موفق میشوند.

## خروجی task2



الگوریتم RM تعداد context switch کمتری دارد.

## خروجی tasks\_interrupts



برای این تسک ست، interrupt ها ابتدا انجام میشوند زیرا اولویت بالاترین را دارند. بین ۳ تسک باقی مانده فرقی نمیکند و همه میتوانند زمانبندی کنند.