

## تمرین سری 6 امبدد

عرفان رفیعی اسکویی – 98243027

پارسا نوری – 98243067

سوال (1)

To check the feasibility of a set of tasks using these algorithms, we need to apply some tests or conditions mentions that for RMS, the necessary and sufficient condition for feasibility is given by:

$$U = \sum_{i=1}^n \frac{C_i}{T_i} \leq n(2^{1/n} - 1)$$

where U is the CPU utilization, n is the number of tasks, C<sub>i</sub> is the computation time of task i, and T<sub>i</sub> is the period of task i.

For EDF, the necessary and sufficient condition for feasibility is given by:

$$U = \sum_{i=1}^n \frac{C_i}{T_i} \leq 1$$

where the symbols have the same meaning as before.

Using these formulas, we can check the feasibility of the given set of tasks:

For RMS :

$$U = \frac{2}{10} + \frac{5}{20} + \frac{8}{30} = 0.7333$$

$$n(2^{1/n} - 1) = 3(2^{1/3} - 1) = 0.7798$$

Since  $U \leq n(2^{1/n} - 1)$ , the set of tasks is **feasible** using RMS.

For EDF:

$$U = \frac{2}{10} + \frac{5}{20} + \frac{8}{30} = 0.7333$$

Since  $U \leq 1$ , the set of tasks is **feasible** using EDF.

Therefore, both RMS and EDF can schedule the given set of tasks without missing any deadlines.

حال برای رسم زمانبندی برای یک فراتناوب به صورت زیر جدول زمانی میکشیم و داریم :

### RMS:

در RMS ، هر فعالیت (task) به یک تناوب (period) ثابت متصل است. تناوب، مدت زمانی است

که یک فعالیت به صورت دوره‌ای تکرار می‌شود. همچنین، هر فعالیت دارای زمان اجرا

(execution time) ثابت است که نشان‌دهنده زمانی است که فعالیت برای اجرای کامل نیاز دارد.

اصل اساسی در RMS این است که فعالیت‌هایی که تناوب کوچک‌تری دارند (به این معنی که نیاز به

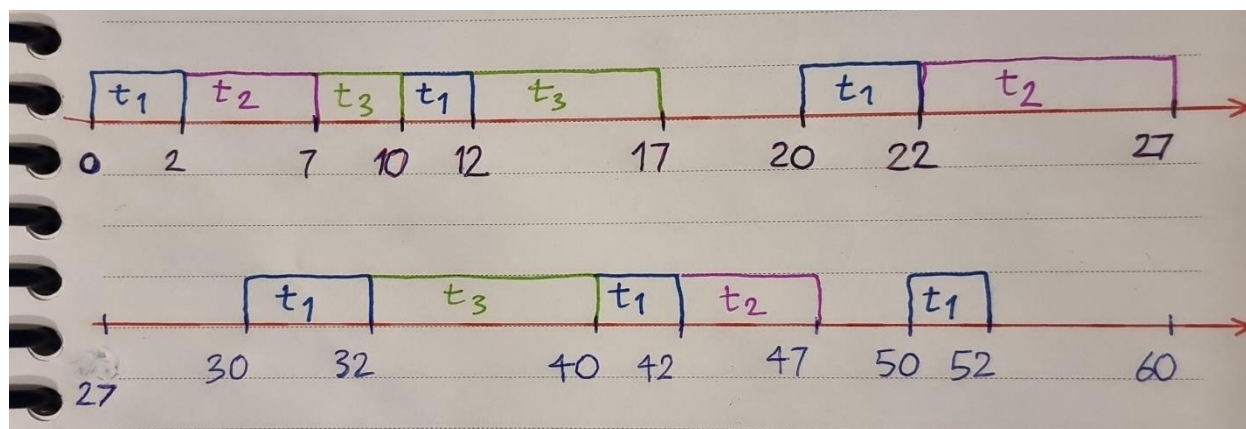
اجرای مکررتری دارند)، باید اولویت بیشتری در زمان‌بندی بگیرند. با داشتن تناوب و زمان اجرا

برای هر فعالیت، می‌توان برای هر فعالیت یک مقدار اولویت (priority) محاسبه کرد.

در RMS، مقدار اولویت هر فعالیت برعکس تناوب آن محاسبه می‌شود. به این معنی که فعالیتی با تناوب کوچک‌تر، اولویت بیشتری دارد. با تعیین مقدار اولویت برای هر فعالیت، الگوریتم زمان‌بندی RMS از روش‌هایی مانند الگوریتم‌های ترتیب غیر صعودی (non-preemptive) یا الگوریتم‌های ترتیب صعودی (preemptive) برای زمان‌بندی فعالیت‌ها استفاده می‌کند.

برای این الگوریتم برای اینکه عملکرد زمان‌بند را بخواهیم بررسی کنیم باید در تایمی به طول ک.م.م دوره تناوب تمامی task ها چک کنیم، چون بعد از این تایم دوباره روی لوپ تکرار می‌فیتیم.

حال در بازه 60 ثانیه داریم :



## EDF:

الگوریتم EDF (Earliest Deadline First) یک الگوریتم زمان‌بندی است که در سیستم‌های عامل واقع شده است. این الگوریتم با توجه به مهلت زمانی (deadline) هر وظیفه، تصمیم می‌گیرد کدام وظیفه را در هر لحظه زمانی اجرا کند. هدف اصلی این الگوریتم، حفظ قابلیت اطمینان سیستم و اجرای به موقع وظیفه‌ها است.

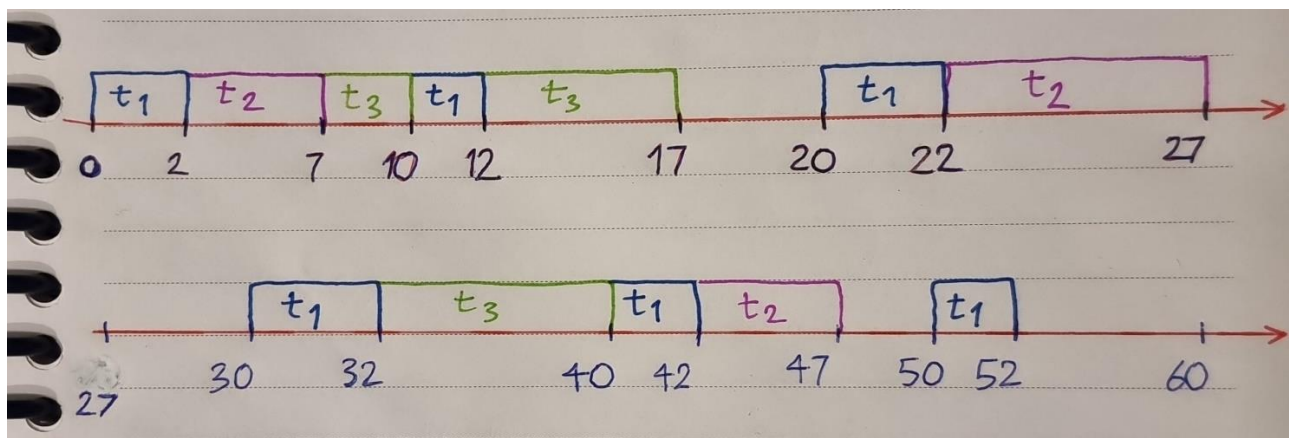
زمانبندی EDF به صورت زیر عمل می‌کند:

1. هر وظیفه دارای مهلت زمانی (deadline) است که نشان دهنده زمانی است که وظیفه باید تا آن زمان به پایان برسد. این مهلت زمانی ممکن است ثابت باشد یا به صورت متغیر در طول زمان تغییر کند.

2. الگوریتم EDF هر لحظه زمانی، وظیفه‌ای را که کوچکترین مهلت زمانی را دارد را برای اجرا انتخاب می‌کند. به این معنی که وظیفه‌ای که مهلت زمانی کمتری دارد، اولویت بیشتری در زمانبندی دارد و اجازه اجرای آن را قبل از وظیفه‌هایی با مهلت زمانی بیشتر می‌دهد.

3. وقتی یک وظیفه اجرا می‌شود، تا زمانی که به پایان برسد یا به خاطر تحریک از بین برود، اجرا خواهد شد و وظیفه‌های دیگری به تأخیر خواهند افتاد. در واقع، EDF بر روی برنامه‌های زمانبندی شده بر اساس سیاست اجرای آنها تأثیر می‌گذارد و اجرا را تا حد ممکن به موقع و طبق مهلت‌های زمانی تعیین شده تنظیم می‌کند.

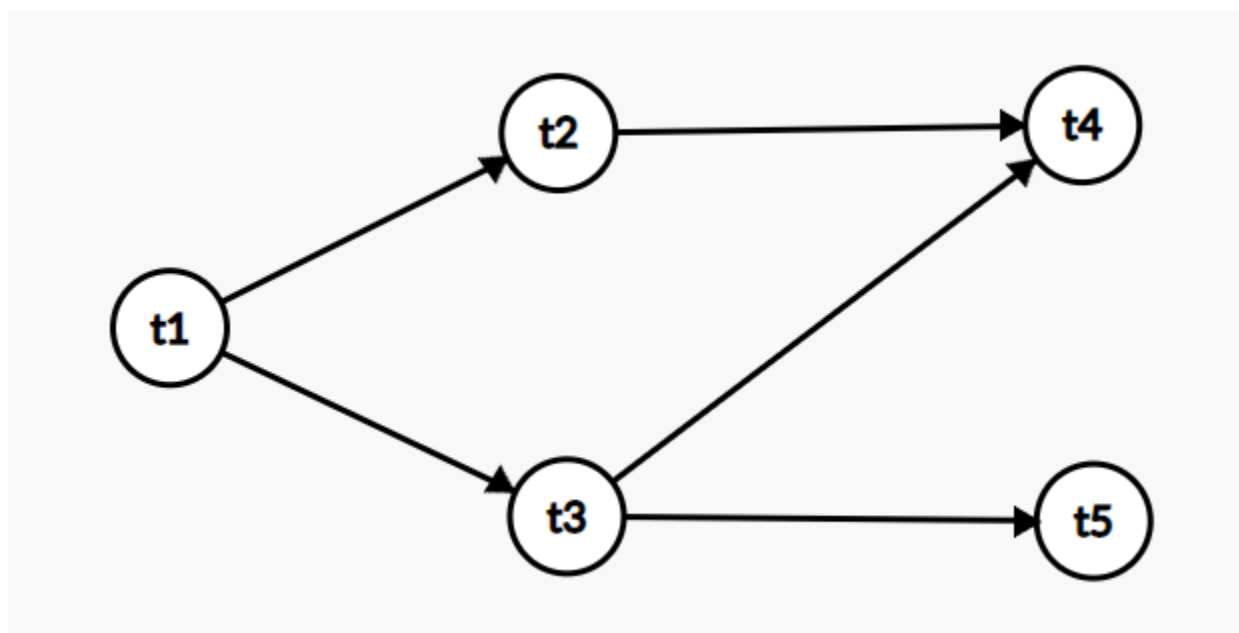
در این سوال، ددلاینی برای وظیفه‌ها مشخص نشده است، پس ددلاین را ساده‌ترین حالت یعنی تا قبل از رسیدن تمام شدن دوره تناوب وظیفه در نظر می‌گیریم و حل ما مثل حالت قبلی یعنی RMS میشود.



سوال 2)

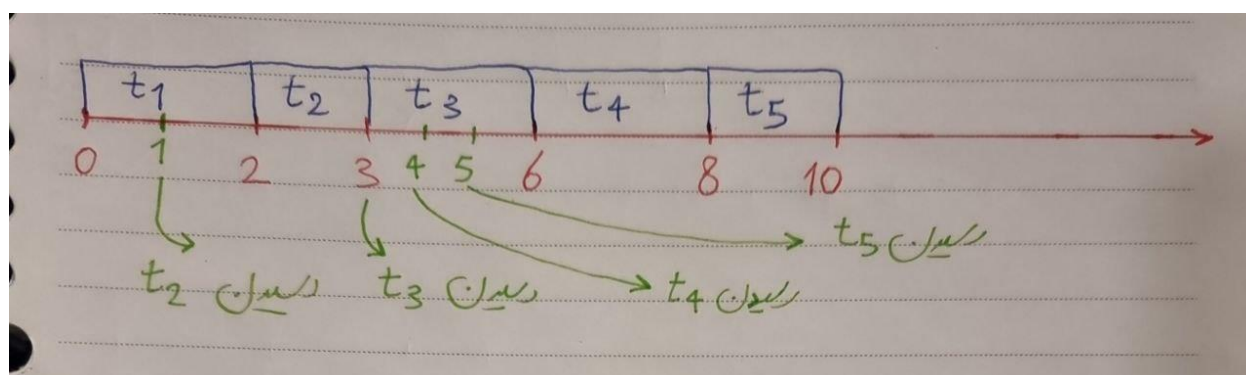
الف)

گراف وابستگی :



ب)

برای الگوریتم EDF داریم :



توضیحات :

ثانیه صفر : در این لحظه، وظیفه یک و ارد میشود و تنها وظیفه موجود می باشد؛ پس زمانبند را در اختیار می گیرد

ثانیه یک : در این لحظه، و وظیفه دو و ارد می شود؛ اما و وظیفه یک در حال اجرا می باشد. از طرفی ددلاین وظیفه یک در لحظه شش می باشد و ددلاین وظیفه دو در لحظه هفت می باشد. البته با توجه به اینکه وظیفه یک، پیشنیاز وظیفه دو میباشد، پس حتما باید وظیفه یک اول اجرا شود و اجرایش تمام شود. پس وظیفه یک به اجرایش ادامه می دهد

ثانیه دو : در این لحظه اجرای وظیفه یک تمام می شود. از طرفی با توجه به اینکه تنها وظیفه موجود، وظیفه دو می باشد، پس شروع به اجرا می کند.

ثانیه سه : در این لحظه وظیفه سه از راه میرسد و اجرای وظیفه دو نیز به اتمام رسیده است. پس وظیفه سه اجرا می شود .

ثانیه چهار : در این لحظه وظیفه چهار از راه می رسد؛ از طرفی در حال اجرای وظیفه سه هستیم. با توجه به اینکه ددلاین وظیفه سه، لحظه هشت می باشد و ددلاین وظیفه چهار لحظه ده می باشد، پس به اجرای وظیفه سه ادامه می دهیم. البته با توجه به اینکه وظیفه سه پیشنیاز وظیفه چهار می باشد، باید صرف نظر از ددلاین ها، به اجرای وظیفه سه ادامه بدهیم.

ثانیه پنج : در این لحظه، وظیفه پنج از راه میرسد؛ اما در حال اجرای وظیفه سه می باشیم. با توجه به اینکه وظایف موجود وظیفه های سه، چهار و پنج هستند ددلاین وظیفه سه از بقیه زودتر می باشد، پس به اجرای وظیفه سه ادامه می دهیم. البته با توجه به اینکه وظیفه سه، پیشنیاز وظایف چهار و پنج می باشد، پس صرف نظر از ددلاین وظیفه های چهار و پنج، به اجرای وظیفه سه ادامه می دهیم .

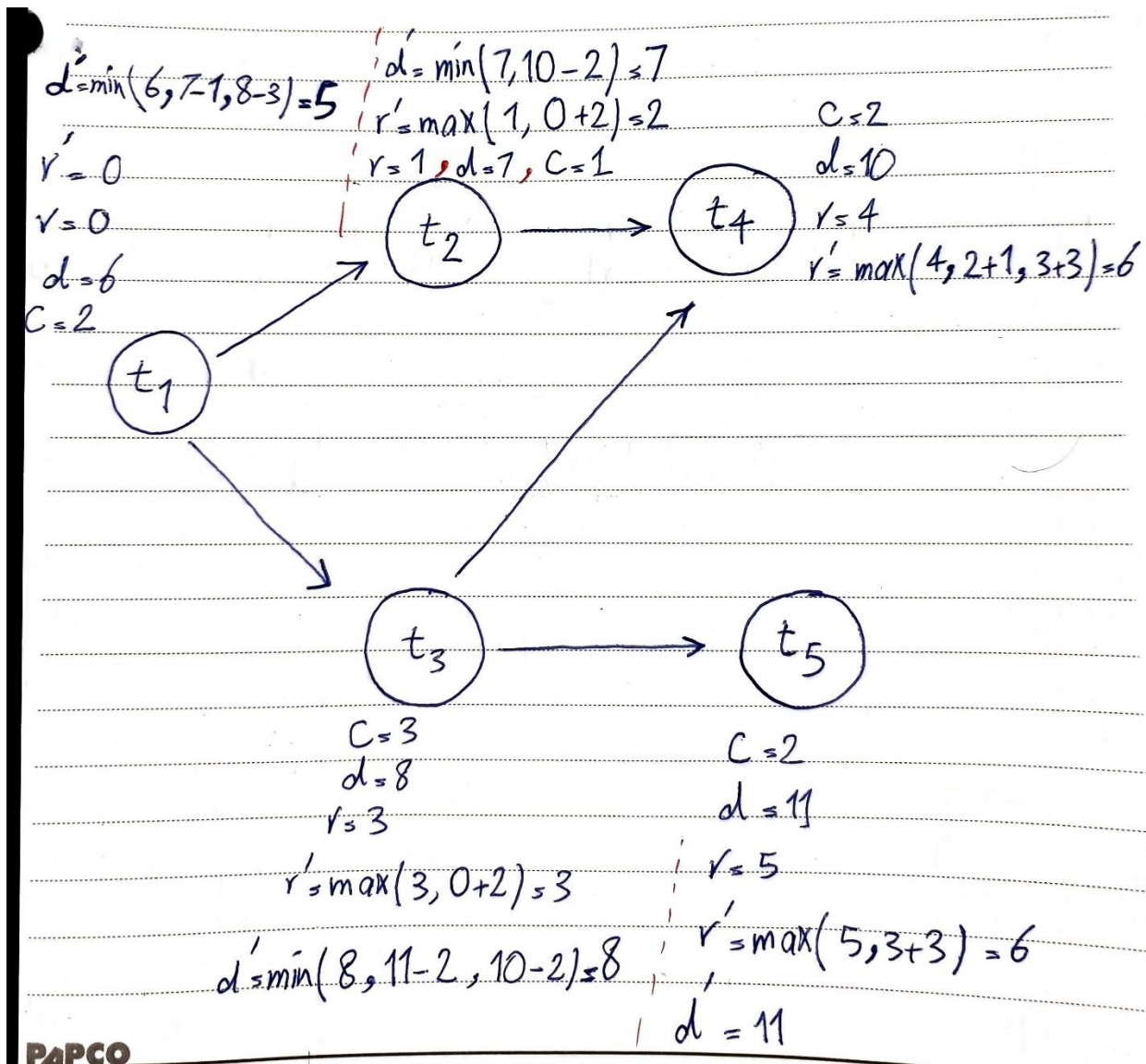
ثانیه شش: در این لحظه، اجرای وظیفه سه به اتمام میرسد. با توجه به اینکه فقط وظایف چهار و پنج موجود هستند و هیچکدام پیشنیاز دیگری نیستند و همه پیشنیاز هایشان اجرا شده اند، پس با توجه به اینکه ددلاین وظیفه چهار لحظه ده می باشد و ددلاین وظیفه پنج لحظه یازده می باشد، پس وظیفه چهار را اجرا می کنیم .

ثانیه هشت: در این لحظه اجرای وظیفه چهار به اتمام میرسد. با توجه به اینکه فقط وظیفه پنج باقی مانده است، پس به اجرای آن می پردازیم .

ثانیه ده: در این لحظه اجرای وظیفه پنج نیز به اتمام میرسد. با توجه به زمانبندی فوق، که در آن تمامی وظایف به ددلاین هایشان رسیدند، پس زمانبندی برای این وظایف، با استفاده از الگوریتم EDF ممکن است.

ج)

این الگوریتم با EDF عادی مشابه است با این تفاوت که قبل از اجرای EDF ما ددلاین ها و زمان های رسیدن را با توجه به نیازمندی ها و پیشنیاز هایمان آپدیت میکنیم. برای مورد ج به روزرسانی EDF\* انجام شده :



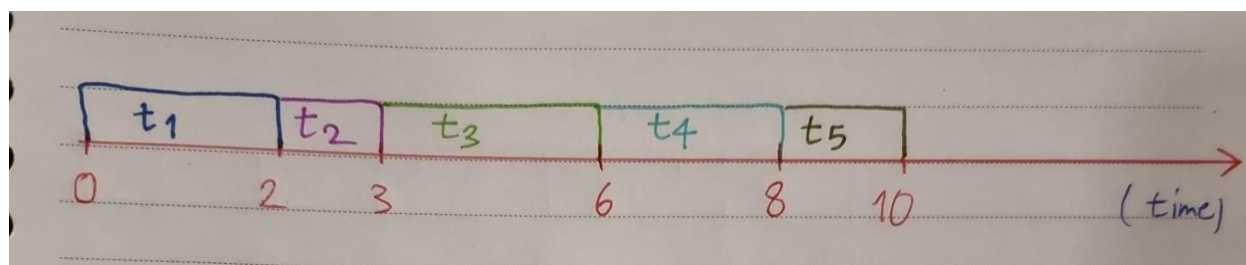
توضیح update ها :

برای بروزرسانی زمان های رسیدن وظایف، هر وظیفه تنها وقتی می تواند شروع شود که پیشنیاز های آن، اجرایشان تمام شده باشد. پس بنابراین باید حداقل مقدار بین زمان ورودی وظایف پیشنیاز آن وظیفه بعلاوه زمانی که طول می کشد تا اجرای آن وظیفه به اتمام برسد و زمان رسیدن آن وظیفه را حساب کرد. این مقدار را برای هر کدام از وظایف حساب کردیم و مقدار آن را در  $r$  قرار دادیم. برای



بروزرسانی ددلاین های وظایف، باید بیایم و ببینیم که وظایفی که این وظیفه پیشنهادشان می باشد، در دیرترین حالت ممکن در چه لحظه ای میتواند شروع به کار کنند و به ددلاینشان برسند. باید بیایم و کمترین مقدار بین این مقادیر و ددلاین خود وظیفه را به عنوان ددلاین بروزرسانی شده معرفی کنیم. این مقدار را محاسبه و در 'd' قرار دادیم.

حال در شکل زیر برای زمانبندی داریم :



توضیح نمودار زمانی :

ثانیه صفر: در این لحظه، وظیفه یک وارد میشود و زمانبند را در اختیار می گیرد و شروع به اجرا می کند .

ثانیه دو: در این لحظه، اجرای وظیفه یک تمام می شود و همچنین با توجه به زمان های ورودی بروزرسانی شده، وظیفه دو نیز وارد می شود. پس زمانبند را در اختیار می گیرد و شروع به اجرا میکند .

ثانیه سه: در این لحظه، اجرای وظیفه دو به اتمام میرسد و همچنین با توجه به زمان های ورودی بروزرسانی شده، وظیفه سه وارد می شود و زمانبند را در اختیار میگیرد و شروع به اجرا می کند.

ثانیه شش: در این لحظه، انجام وظیفه سه به پایان میرسد. سپس با توجه به زمان های ورود  
بروزرسانی شده، وظیفه های چهار و پنج وارد می شوند. با توجه به اینکه الگوریتم ما EDF می  
باشد، پس با توجه به نزدیک تر بودن ددلاین بروزرسانی شده وظایف، وظیفه چهار زمانبند را در  
اختیار می گیرد و ش روه به اجرا می کند.

ثانیه هشت: در این لحظه، اجرای وظیفه چهار به پایان میرسد و با توجه به اینکه وظیفه پنج باقی  
مانده است، پس زمانبند را در اختیار میگیرد و شروه به اجرا می کند.

ثانیه ده: در این لحظه اجرای وظیفه پنج نیز به اتمام میرسد.

پس دیدیم که feasible است.