

تحلیل زمان بندی بی درنگ با استفاده از دو الگوریتم RM و EDF

استاد امین عنایت زارع

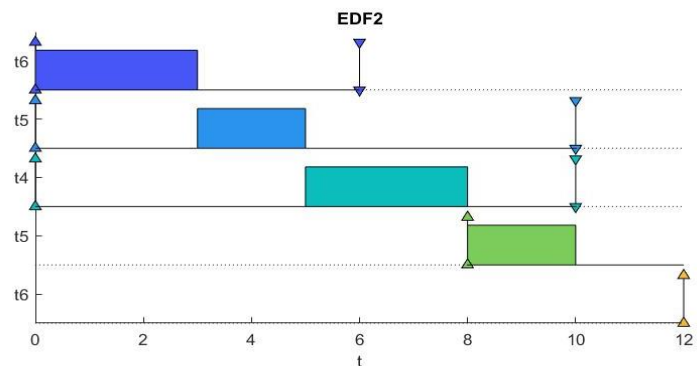
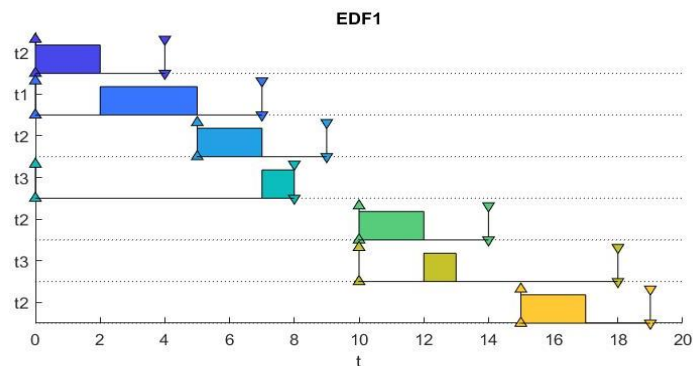
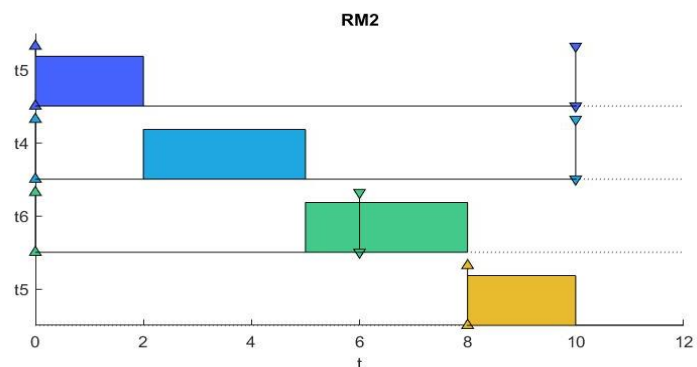
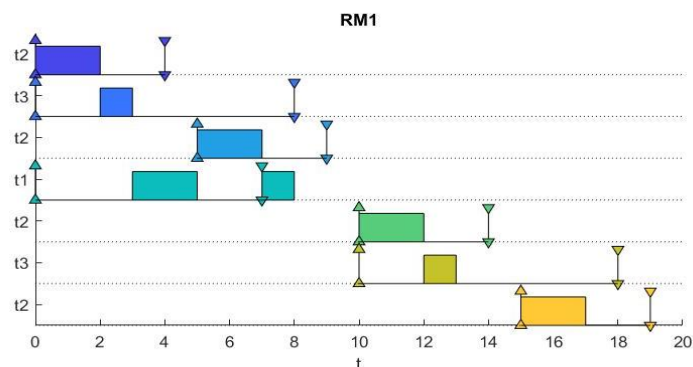
اعضای گروه:

محمد حسین موذن نیا

زینب واعظزاده

صادق همدانی پور

شرح مجموعه وظایف



مجموعه سری اول برای تحلیل و بررسی انتخاب شده. مجموعه و زمان بندی سری دوم نیز مشابه سری اول خواهد بود و نتیجه گیری یکسان. مجموعه اول وظیفه های ما در اینجا از سه وظیفه t_1 ، t_2 و t_3 تشکیل شده است.

وظیفه t_1 دارای زمان اجرای ۳، دوره تناوب ۲، زمان ورود ۰ و مهلت اجرای ۷ می باشد.

وظیفه t_2 نیز زمان اجرای ۲، دوره تناوب ۵، زمان ورود ۰ و مهلت اجرای ۴ را دارد.

در نهایت t3 را داریم که پارامترهای آن نیز شامل زمان اجرای ۱، دوره تناوب ۱۰، زمان ورود ۰ و مهلت اجرای ۸ است.

تحلیل نحوه اجرای الگوریتم زمان بندی RM

همانگونه که می دانیم در این الگوریتم اولویت هر وظیفه براساس نرخ آن، یعنی معکوس دوره تناوب آن وظیفه تعیین می شود. در نتیجه بالاترین اولویت به وظیفه ای تخصیص داده می شود که کمترین دوره تناوب را دارد.

در اینجا نرخ هر سه وظیفه تعریف شده را محاسبه می کنیم.

$$R = \frac{1}{T}$$

$$R_1 = \frac{1}{20} = 0.05$$

$$R_2 = \frac{1}{5} = 0.2$$

$$R_3 = \frac{1}{10} = 0.1$$

از محاسبه نرخ وظیفه ها متوجه می شویم بالاترین اولویت با t_2 بوده و پس از آن t_3 و در نهایت آخرین اولویت با t_1 می باشد.

قبل از اجرای زمان بندی شرط لازم و کافی الگوریتم را بررسی می کنیم.

شرط لازم:

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

$$\frac{C_1}{T_1} + \frac{C_2}{T_2} + \frac{C_3}{T_3} \leq 1$$

$$\frac{3}{20} + \frac{2}{5} + \frac{1}{10} = \frac{13}{20} = 0.65$$

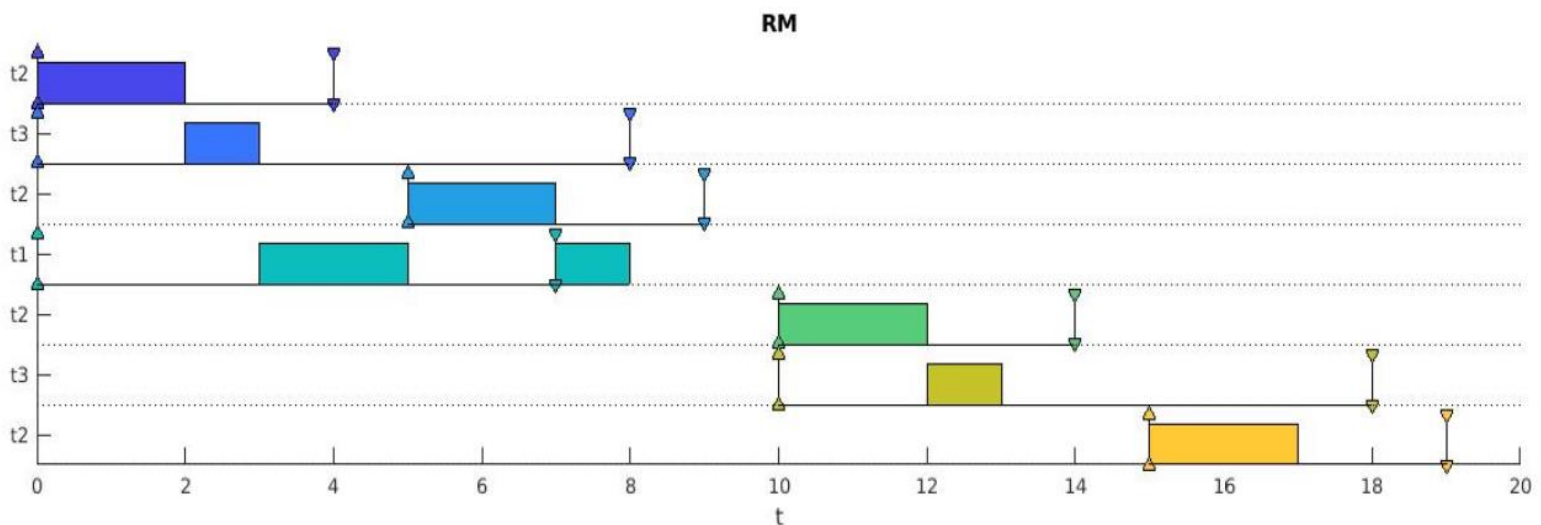
$$0.65 \leq 1 \checkmark$$

شرط کافی:

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

$$0.65 \leq 0.779 \checkmark$$

با انجام محاسبات بالا متوجه می‌شویم که این مجموعه وظایف هم شرط لازم و هم شرط کافی زمان‌بندی RM را دارند.



تصویر بالا نمودار زمان‌بندی این وظیفه‌ها با الگوریتم RM می‌باشد.

همانگونه که انتظار می‌رفت اول t_2 اجرا شده (از لحظه ۰ تا ۲)، سپس چون اولویت بعدی با t_3 است، از لحظه ۲ تا ۳ t_3 اجرا شده است. بعد از آن نوبت t_1 شده و این وظیفه پردازنده را در اختیار می‌گیرد ولی پس از گذر ۲ واحد زمانی (در لحظه ۵) و قبل از اتمام آن، وظیفه t_2 دوباره وارد سیستم می‌شود

و چون الگوریتم از نوع قبضه ای و غیرانحصاری است، سیستم عامل پردازنده را از t_1 گرفته و اختیار t_2 قرار می دهد. پس از ۲ واحد زمانی و اتمام وظیفه t_2 ، سیستم عامل دوباره پردازنده را در اختیار t_1 قرار می دهد تا به اتمام برسد. ولی مهلت زمانی t_1 که لحظه ۷ بوده بلافاصله پس از اتمام وظیفه t_2 سر می رسد. در نتیجه وظیفه t_1 به اصطلاح miss می شود.

تا لحظه ۱۰ که وظیفه های t_2 و t_3 وارد سیستم می شوند، زمان بیکاری پردازنده بوده (Idle Time)، از ۱۰ تا ۱۲ پردازنده در اختیار t_2 بوده و پس از اتمام آن t_3 پردازنده را در اختیار می گیرد. پس از اتمام t_3 یعنی لحظه ۱۳ هیچ وظیفه ای در صف آماده نیست پس پردازنده دوباره وارد زمان بیکاری می شود. تا اینکه در لحظه ۱۵ دوره تناوب t_2 سر می رسد و این وظیفه وارد سیستم می شود. بعد از تمام شدن t_2 باز هم صف آماده خالی شده و هیچ وظیفه ای برای انجام وجود ندارد در نتیجه پردازنده وارد زمان بیکاری می شود.

در لحظه ۲۰ همه وظیفه ها به صورت همزمان دوباره وارد سیستم شده و تمام سناریو گفته شده در بالا تکرار می شود.

تحلیل نحوه اجرای الگوریتم زمان بندی EDF

در الگوریتم EDF که یکی از الگوریتم های زمان بندی کلاس پویا می باشد، اولویت هر وظیفه بر اساس مهلت زمانی آن وظیفه تعیین می شود. در نتیجه بالاترین اولویت به وظیفه ای داده می شود که کوتاه ترین مهلت زمانی مطلق را دارد.

در الگوریتم EDF شرط لازم و کافی برای قابل زمان بندی بودن یک وظیفه دوره ای این است که مهلت زمانی مطلق آن وظیفه با دوره تناوب آن برابر باشد.

یک مجموعه وظیفه دوره ای که در آن مهلت زمانی مطلق با دوره تناوب برابر است اگر و تنها اگر عامل بهره وری پردازنده (PUF) کوچک تر یا مساوی یک باشد، یعنی:

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

$$U_1 = \frac{3}{20} = 0.15$$

$$U_2 = \frac{2}{5} = 0.4$$

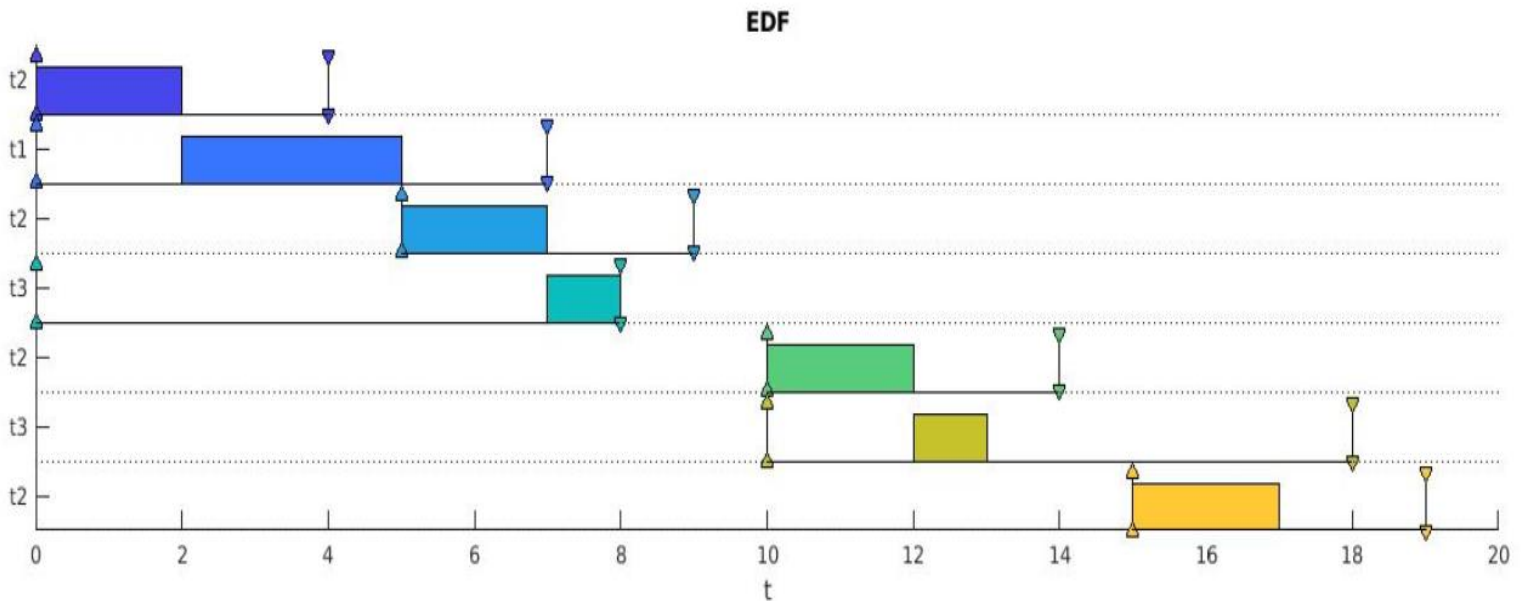
$$U_3 = \frac{1}{10} = 0.1$$

$$U_1 + U_2 + U_3 \leq 1$$

$$0.15 + 0.4 + 0.1 = 0.65$$

$$0.65 \leq 1 \checkmark$$

در نتیجه این مجموعه وظایف با الگوریتم EDF قابل زمان بندی می باشند.



طبق تعریف وظیفه ها و در الگوریتم EDF بالاترین اولویت با t2 بوده و بعد از آن t1 و در آخر t3.

همانگونه که در تصویر بالا می بینیم اول t2 اجرا شده (از لحظه ۰ تا ۲)، سپس t1 از لحظه ۲ تا ۵ اجرا شده است. بعد از اتمام t1 وظیفه t2 دوباره وارد سیستم شده و در صف آماده قرار می گیرد. الان دو وظیفه t2 و t3 در صف آماده قرار دارند ولی چون اولویت t2 بالاتر است، این وظیفه پردازنده را در اختیار می گیرد.

بعد از اتمام t2 تنها t3 در صف آماده وجود دارد پس سیستم عامل پردازنده را به آن می دهد. t3 دقیقا در آخرین لحظه مهلت اجرای خود به اتمام می رسد.

بعد از تمام شدن t_3 هیچ وظیفه‌ای در صف اولویت وجود ندارد و پردازنده وارد زمان بیکاری (Idle Time) می‌شود. تا اینکه در لحظه ۱۰ هر دو وظیفه t_2 و t_3 وارد سیستم می‌شوند. با توجه به اولویت بالاتر t_2 اول این وظیفه اجرا می‌شود و پس از اتمام آن t_3 پردازنده را در اختیار می‌گیرد. وظیفه t_3 در لحظه ۱۳ به اتمام رسیده و تا لحظه ۱۵ که وظیفه t_2 وارد سیستم می‌شود پردازنده در زمان بیکاری است. بعد از ورود t_2 در لحظه ۱۵ سیستم عامل پردازنده را در اختیار آن قرار می‌دهد. برای تکمیل t_2 به ۲ واحد زمانی نیاز داریم. پس از تکمیل اجرای t_2 هیچ وظیفه‌ای در صف آماده وجود ندارد و پردازنده دوباره وارد زمان بیکاری می‌شود.

در نهایت در لحظه ۲۰ همه وظیفه وارد سیستم شده و تمام سناریو گفته شده در بالا دوباره تکرار می‌شود.

نتیجه‌گیری و مقایسه

پس از اجرای دو الگوریتم روی دو مجموعه وظیفه به نتایج زیر می‌رسیم:

- ❖ این مجموعه وظایف شرط لازم و کافی الگوریتم RM را دارند.
- ❖ در مجموعه اول الگوریتم RM بالاترین اولویت با t_2 بوده و پس از آن t_3 و در نهایت t_1 .
- ❖ در مجموعه اول الگوریتم RM و در سیکل اول اجرای آن، وظیفه t_1 miss شده و در مهلت مقرر خود به اتمام نمی‌رسد.
- ❖ دلیل این اتفاق قبضه‌ای بودن این الگوریتم است.
- ❖ مجموع زمان بیکاری پردازنده (Idle Time) در یک سیکل ۲۰ واحدی الگوریتم RM ۷ واحد زمانی است.
- ❖ این مجموعه وظایف شرط لازم و کافی الگوریتم EDF را هم دارند.
- ❖ در مجموعه اول الگوریتم EDF بالاترین اولویت با t_2 بوده و پس از آن t_1 و در نهایت t_3 .
- ❖ مجموع زمان بیکاری پردازنده (Idle Time) در یک سیکل ۲۰ واحدی الگوریتم EDF ۷ واحد زمانی است.
- ❖ در مجموعه دوم الگوریتم RM وظیفه t_6 ددلاین خود را از دست داده و miss می‌شود. درحالی که الگوریتم EDF به درستی زمان‌بندی را انجام می‌دهد.
- ❖ در الگوریتم EDF همه وظیفه‌ها در هر دو مجموعه در مهلت زمانی خود به اتمام می‌رسند و هیچ وظیفه‌ای miss نمی‌شود.

❖ در نهایت می‌توان نتیجه گرفت که در اجرای این دو مجموعه وظیفه الگوریتم EDF بهتر عمل می‌کند.