



پروژه‌های درس سیستم‌های بی‌درنگ

مدرس: دکتر صفری

دانشجویان محترم لیست پروژه‌های درس سیستم‌های بی‌درنگ در ادامه آورده شده‌است. هر گروه (یک نفره یا دو نفره) باید ۵ اولویت اول خود از بین پروژه‌های زیر را انتخاب کند. شما می‌توانید با دستیار آموزشی هر پروژه در طول ترم در ارتباط باشید و پروژه را پیاده‌سازی کنید.

## عنوان: زمان‌بندی وظایف و مدیریت منابع به صورت بی‌درنگ برای سیستم‌های پهنپای چند

دستیار آموزشی: آقای یونسی

### هسته‌ای

**توضیحات:** در این پروژه، ما به زمان‌بندی وظایف و مدیریت منابع به صورت بی‌درنگ در زمینه وسایل نقلیه هوایی بدون سرنشین (پهنپایها) ۸ هسته ای می‌پردازیم. هدف این پروژه طراحی و پیاده‌سازی یک الگوریتم زمان‌بندی (تغییر پویای ولتاژ و فرکانس آگاه از انرژی<sup>۱</sup>) برای اجرای وظایف پهنپای بر روی یک پلتفرم چند هسته‌ای است. ما همچنین از تکنیک مدیریت منابع (مدیریت مبتنی بر کنترل بازخورد<sup>۲</sup>) جهت استفاده کارآمد از منابع موجود بهره خواهیم برد و در نتیجه عملکرد سیستم پهنپای را بهینه می‌کنیم. برای ارائه یک تجربه عملی با استفاده از الگوریتم JIJUNIFAST، مجموعه‌ای جامع از وظایف نرم و سخت پهنپایها (برای مثال جدول زیر را مشاهده کنید) را همراه با ویژگی‌های زمان‌بندی آنها باید تولید کنید.

جدول ۱ ( نمونه وظایف با بهره‌وری ۰.۴)

Task	Computation Time	Deadline	Criticality	Utilization
T1	12 ms	25 ms	Soft	0.02
T2	6 ms	10 ms	Soft	0.08
T3	20 ms	30 ms	Hard	0.012
T4	8 ms	15 ms	Hard	0.1
T5	10 ms	12 ms	Soft	0.078
T6	18 ms	20 ms	Hard	0.11
T7	14 ms	25 ms	Soft	0.09

خروجی مورد نظر بر اساس بهره‌وری های 0.3, 0.5, 0.7:

- نمودار زمان‌بندی پذیری و نرخ وظایف انجام نشده
- نمودار makespan
- نمودار میانگین زمان انتظار و لختی
- نمودار ترتیب اجرای وظایف بر روی هسته‌ها

فاز اول:

- تولید ۱۰۰ وظیفه توسط الگوریتم معرفی شده
- پیاده‌سازی زمان‌بندی سیستم ۸ هسته‌ای
  - نمودار زمان‌بندی پذیری و نرخ وظایف انجام نشده
  - نمودار ترتیب اجرای وظایف بر روی هسته‌ها

فاز دوم:

- پیاده‌سازی الگوریتم مدیریت منابع
  - تمامی نمودارها در گزارش اضافه شوند.

<sup>1</sup> Energy-Aware Dynamic Voltage and Frequency Scaling (EDVFS)

<sup>2</sup> Feedback-based Control Management

یک پهپاد امدادگر وظایف ارسال دارو و غذا و تصویربرداری از یک گروه گرفتار در کوهستان را دارد. وظیفه اول این پهپاد یک وظیفه حیاتی است و وظیفه دوم یک وظیفه نرم که انجام نشدن آن به صورت کامل خطر جانی‌ای برای کوهنوردان ندارد. با توجه به امکان وجود اشکال و حیاتی بودن وظایف نیاز است تا از روش تحمل‌پذیری اشکال تکرار وظایف (Task Replication) برای مقابله با اشکال برای وظایف حیاتی استفاده کنیم. شما بعنوان یک مهندس سامانه‌های بی‌درنگ باید یک الگوریتم زمانبندی برای اجرای وظایف بر روی سامانه بحرانی-مختلط این پهپاد امدادگر پیاده‌سازی کنید که بصورت زیر باشد:

سامانه بحرانی-مختلط چند هسته‌ای شامل وظایف پرریودیک (متناوب) LC و HC می‌باشد که وظایف LC دارای یک زمان اجرا هستند در حالی که وظایف HC دارای دو بدترین زمان اجرا هستند. ابتدا با توجه به اینکه قابلیت اطمینان سامانه حداقل ۰.۹۹۹۹۹۹ باشد تعداد کپی لازم برای هر یک از وظایف HC را مشخص کنید. سپس وظایف را براساس دو سیاست Worst Fit Decreasing (WFD) و First Fit Decreasing (FFD) بر روی هسته‌ها نگاشت کنید. توجه کنید که اولویت با نگاشت وظایف اصلی و کپی‌های آن است. سپس وظایف LC باید نگاشت و زمانبندی شوند. سپس باید وظایف براساس الگوریتم زمانبندی ER-EDF بر روی هسته‌ها زمانبندی شوند. سامانه در حالت نرمال شروع به کار می‌کند و در صورتی که یکی از وظایف HC به اندازه زمان اجرای کوچک خود اجرا شود ولی به پایان نرسد سامانه وارد حالت overrun می‌شود. توجه نمایید که اجرای درست و به موقع وظایف HC برای سامانه حیاتی است در حالی که نیاز است تا یک حداقل کیفیت خدمات برای سامانه تضمین شود. در نهایت نتایج زیر را باید در گزارش خود ثبت و ارایه کنید:

الف) حالتی که همه هسته‌ها در حالت نرمال کار کنند:

۱. نمودار قابلیت زمانبندی برای سامانه ۸ هسته‌ای با نسبت وظایف HC به LC برابر یک و با بهره‌وری ۰.۵.
۲. نمودار کیفیت خدمات برای سامانه ۸ هسته‌ای با نسبت وظایف HC به LC برابر یک و با بهره‌وری ۰.۵.
۳. زمانبندی انجام شده برای سامانه ۸ هسته‌ای با نسبت وظایف HC به LC برابر یک و با بهره‌وری ۰.۵.

ب) حالتی که ۳۰ درصد هسته‌ها دچار اشکال و ۵۰ درصد هسته‌ها دچار overrun شوند:

۱. نمودار قابلیت زمانبندی برای سامانه ۱۶ هسته‌ای با نسبت وظایف HC به LC برابر یک و با بهره‌وری ۰.۷۵.
۲. نمودار کیفیت خدمات برای سامانه ۱۶ هسته‌ای با نسبت وظایف HC به LC برابر یک و با بهره‌وری ۰.۷۵.
۳. زمانبندی انجام شده برای سامانه ۱۶ هسته‌ای با نسبت وظایف HC به LC برابر یک و با بهره‌وری ۰.۷۵.

در نهایت همچنین باید وظایف را با استفاده از الگوریتم EDF نیز زمانبندی کنید و نتایج ذکر شده در بالا را برای این الگوریتم نیز به دست آورید و بررسی کنید که چرا این الگوریتم برای سامانه‌های بحرانی-مختلط مناسب نیست.

فازبندی پروژه:

۱. شما باید در فاز اول به وسیله الگوریتم UUnifast تعدادی وظیفه بحرانی-مختلط تولید کنید و سپس تعداد کپی برای هر وظیفه را مشخص کرده و وظایف را بر روی هسته‌ها نگاشت کنید. (باید نتیجه نگاشت وظایف به هسته‌ها در سامانه ۸ هسته‌ای و ۱۶ هسته‌ای با بهره‌وری های ۰.۵ و ۰.۷۵ را در فاز اول ارائه دهید.)

۲. در فاز نهایی تمام نتایج باید گزارش شوند.

سامانه مدیریت پرواز، یک سامانه بحرانی-مختلط است که دارای وظایف با درجه بحرانی بالا مانند اجتناب از برخورد و وظایف با درجه بحرانی پایین مانند کنترل دمای کابین است. اجرا نشدن صحیح وظایف با درجه بحرانی پایین، خطر جانی در پی ندارد و فقط کیفیت خدمات را کاهش می‌دهد اما اجرا نشدن صحیح وظایف با درجه بحرانی بالا باعث ایجاد فاجعه می‌شود. در نتیجه، سامانه بحرانی-مختلط دوسطحی که نوعی از سامانه‌های بحرانی-مختلط است شامل وظایف دوره‌ای LC و HC می‌باشد که وظایف LC دارای یک بدترین زمان اجرا و وظایف HC دارای دو بدترین زمان اجرا هستند. سامانه در حالت نرمال شروع به کار می‌کند و در صورتی که یکی از وظایف HC زمان اجرای کوچک خود را رد کند سامانه وارد حالت overrun می‌شود که در این حالت وظایف با درجه بحرانی بالا با زمان اجرای بزرگ خود اجرا می‌شوند.

شما به عنوان یک مهندس سامانه‌های بی‌درنگ باید وظایف موردنظر را تحت پروتکل Stack Resource Policy و توسط الگوریتم زمانبندی ER-EDF با در نظر گرفتن چند واحدی بودن منابع مشترک زمانبندی کنید و در راستای این پیاده‌سازی، منابع باید به گونه‌ای به وظایف اختصاص داده شوند که هم دسترسی تودرتو و هم سریالی به منابع وجود داشته باشد. علاوه بر استفاده از مجموعه وظایف موجود تحت عنوان وظایف FMS، شما باید به کمک الگوریتم Uunifast مجموعه‌ای از وظایف مصنوعی نیز تولید کنید و درستی پیاده‌سازی خود را گزارش کنید. با توجه به پارامترهای گفته شده در ادامه، نتایج را ثبت و گزارش کنید.

نمودارهای زمانبندی پذیری بر اساس بهره‌وری های ۰.۳، ۰.۵ و ۰.۷۵:

- تولید ۱۰۰ وظیفه با نسبت تعداد وظایف HC به LC برابر یک، تعداد واحد هر منبع یک مقدار تصادفی بین ۱ تا ۵، تعداد کل منابع برابر با ۱۰ و تعداد بخش‌های بحرانی در هر وظیفه، یک مقدار تصادفی بین ۰ تا ۸.
- تولید ۱۰۰ وظیفه با نسبت تعداد وظایف HC به LC برابر یک، تعداد واحد هر منبع یک مقدار تصادفی بین ۱ تا ۵، تعداد کل منابع برابر با ۱۵ و تعداد بخش‌های بحرانی در هر وظیفه، یک مقدار تصادفی بین ۶ تا ۱۰.

نمودارهای کیفیت خدمات بر اساس بهره‌وری های ۰.۵، ۰.۷ و ۰.۹:

- تولید ۱۰۰ وظیفه با نسبت تعداد وظایف HC به LC برابر یک، تعداد واحد هر منبع یک مقدار تصادفی بین ۱ تا ۵، تعداد کل منابع برابر با ۱۰ و تعداد بخش‌های بحرانی در هر وظیفه، یک مقدار تصادفی بین ۰ تا ۸.
- تولید ۱۰۰ وظیفه با نسبت تعداد وظایف HC به LC برابر یک، تعداد واحد هر منبع یک مقدار تصادفی بین ۱ تا ۵، تعداد کل منابع برابر با ۱۵ و تعداد بخش‌های بحرانی در هر وظیفه، یک مقدار تصادفی بین ۶ تا ۱۰.

فازبندی پروژه:

۱. فاز اول: در فاز اول وظایف تولید شده، پیاده‌سازی الگوریتم پیشنهادی برای تولید و نگاشت منابع و تخصیص بخش بحرانی به وظایف و تعیین سطوح پیش‌دستی به عنوان نتایج خروجی گزارش شوند.
۲. فاز دوم: در فاز نهایی، تمام نمودارهای خواسته شده گزارش شوند.

در این پروژه به زمان‌بندی وظایف در خودروهای امروزی می‌پردازیم. امروزه در خودروها ECU وظیفه حفظ، مدیریت و کنترل خودرو را بر عهده دارد. این قطعه عضو حیاتی خودروها است و عملکردهایی همچون تزریق سوخت، انجام عملیات کاربر (چرخاندن فرمان، گرفتن ترمز و ...)، دریافت اطلاعات از سنسورها و تحلیل آن (پیشبینی برخورد در برخی خودروها) را انجام می‌دهد. بدیهی است که برخی از این عملکردها مانند پیشبینی برخورد و گرفتن ترمز از اهمیت بیشتری برخوردار هستند و برخی مانند خواندن دما و نمایش آن بر روی مانیتور اهمیت کمتری دارند. هدف این پروژه طراحی و پیاده‌سازی الگوریتم‌هایی برای زمان‌بندی عملکردهای خودرو است. سه الگوریتم FDWS، FDS\_MIMF و ADS\_MIMF برای این منظور ارائه شده‌اند. این سه الگوریتم مشابه را پیاده‌سازی کرده و آن‌ها را با یکدیگر مقایسه نمایید.

محدودیت‌های پیاده‌سازی بدین صورت هستند: (۱) در این سیستم هر عملکرد دارای یک سطح بحرانی و یک مهلت زمانی است و شامل مجموعه‌ای از وظایف است که به صورت گراف وظایف (DAG) مدل‌سازی می‌شوند. (۲) عملکردها (گراف‌ها) به صورت پویا وارد سیستم می‌شوند. (۳) عملکردها دارای سطوح بحرانی کم یا زیاد هستند. (۴) سیستم چندهسته‌ای شامل هسته‌های ناهمگن است. (۵) هر عملکرد (گراف) دارای یک lower bound است که از طریق الگوریتم معروف HEFT بدست می‌آید.

شبیه‌سازی:

- وظایف هر عملکرد را به دو صورت Gaussian Elimination و Fast Fourier Transformation تولید کنید (تعداد وظایف را بیش از 220 وظیفه در نظر بگیرید).
- سطح بحرانی هر عملکرد را به صورت تصادفی و به کمک یک احتمال (احتمال بحرانی زیاد بودن) تولید کنید.
- زمان اجرای وظایف در بازه [10, 100] و هزینه ارتباطی را در بازه [5, 25] به صورت یکنواخت تولید کنید.
- از آنجاکه سیستم ناهمگن است، زمان اجرای وظایف بر روی هر پردازنده و هزینه ارتباطی بین دو وظیفه بر روی هر دو پردازنده باید به صورت تصادفی در بازه گفته شده تولید گردد.
- زمان رسیدن عملکردها (گراف‌ها) را به صورت تصادفی و در بازه [0, 1000] انتخاب کنید.

خروجی‌های مورد نظر:

برای ۳ حالت که در آن مهلت زمانی هر عملکرد ۱.۱، ۱.۲ و ۱.۵ برابر lower bound آن است، نمودارهای زیر را تولید کنید (۱۲ نمودار):

- نمودار makespan نهایی سیستم بر حسب تعداد عملکردها: برای هر سه الگوریتم گفته شده این نمودار را برای ۴، ۸، ۱۶ و ۳۲ عملکرد که در یک سیستم با ۸ هسته پردازشی اجرا می‌شوند، تولید کنید.
- نمودار makespan نهایی سیستم بر حسب تعداد هسته‌های پردازشی: برای هر سه الگوریتم گفته شده این نمودار را برای ۴، ۸، ۱۶ و ۳۲ عملکرد که در یک سیستم با ۸ هسته پردازشی اجرا می‌شوند، تولید کنید.
- نمودار DMR نهایی سیستم بر حسب تعداد عملکردها: برای هر سه الگوریتم گفته شده این نمودار را برای ۴، ۸، ۱۶ و ۳۲ عملکرد که در یک سیستم با ۸ هسته پردازشی اجرا می‌شوند، تولید کنید.
- نمودار DMR عملکردها با سطوح بحرانی بالا بر حسب تعداد عملکردها: برای هر سه الگوریتم گفته شده این نمودار را برای ۴، ۸، ۱۶ و ۳۲ عملکرد که در یک سیستم با ۸ هسته پردازشی اجرا می‌شوند، تولید کنید.

فاز ۱:

- پیاده‌سازی الگوریتم تولید وظایف و تولید ۸ عملکرد با آن.
- پیاده‌سازی الگوریتم HEFT و محاسبه lower bound هر عملکرد.

فاز ۲:

پیاده‌سازی تمامی موارد گفته شده، تولید تمام خروجی‌های خواسته شده و تهیه گزارش پایانی.

**توضیحات:** در این پروژه، به زمانبندی وظایف متناوب با هدف کاهش Deadline Miss-rate و انرژی مصرفی در سیستم‌های چند هسته‌ای همگن Non-preemptive است. در این سیستم می‌توان با اختصاص دادن تعداد بیشتری هسته زمان اجرای وظیفه را کاهش داد، اما پس از اختصاص یک هسته به یک وظیفه تا زمان پایان آن وظیفه، نمی‌توان هسته را از آن گرفت. همچنین فرض می‌کنیم قابلیت اجرای همزمان حداکثر دو وظیفه وجود دارد و همه هسته‌ها باید بین دو وظیفه تقسیم شوند و یا بیکار بمانند.

### الگوریتم Cooperative

در این الگوریتم به ترتیب وظایف را اجرا و به هر کدام از آنها هسته‌ها را اختصاص می‌دهیم. این الگوریتم به عنوان Baseline در نظر گرفته شود.

### الگوریتم sBEET

در این الگوریتم فرض می‌کنیم قابلیت اجرای همزمان حداکثر دو وظیفه وجود دارد. وظایف را به ترتیب اولویت (که در اینجا مشابه الگوریتم RM است) مرتب می‌کنیم. سپس اولین وظیفه را انتخاب می‌کنیم و به ازای همه حالات اختصاص هسته‌ها به آن، زمانبندی‌ها را در نظر می‌گیریم. سپس بین این زمانبندی‌ها آن‌هایی که منجر به از دست رفتن موعد زمانی (ددلاین) سایر وظایف در آینده نمی‌شوند را انتخاب می‌کنیم. اگر چندین زمانبندی از این نظر مشابه بودند بین آن‌ها زمانبندی با کمترین انرژی مصرفی را انتخاب می‌کنیم. به این ترتیب ابتدا Deadline Miss-rate و سپس انرژی مصرفی را کاهش می‌دهیم.

ورودی‌ها:

- پروفایل شامل زمان اجرا، توان، و انرژی مصرفی هر وظیفه به ازای تعداد هسته‌ها (حداکثر ۶ هسته)
- مجموعه وظایف شامل دوره تناوب و Utilization هر وظیفه

خروجی‌ها:

- نمودار زمانبندی الگوریتم‌ها
- نمودار توان الگوریتم‌ها
- نمودار مقایسه Deadline Miss-rate با افزایش Utilization
- نمودار مقایسه انرژی مصرفی با افزایش Utilization
- نمودار Speedup نسبت به الگوریتم Cooperative با افزایش Utilization

فاز اول:

- نمودار زمانبندی الگوریتم‌ها

فاز دوم:

- سایر نمودارها و تمام خروجی‌های مورد نیاز

## عنوان: زمان‌بندی هوشمند وظایف برای موبایل با انرژی کارآمد در دستگاه‌های لبه چند هسته‌ای

دستیار آموزشی: آقای یونسی

**توضیحات:** این پروژه با استفاده از رویکرد نوآورانه الهام گرفته از Levy-walk، بر طراحی یک الگوریتم زمان‌بندی پیشرفته تمرکز دارد که به طور هوشمند وظایف را برای به حداکثر رساندن بهره‌وری انرژی و در عین حال رعایت محدودیت‌های بی‌درنگ اختصاص می‌دهد. علاوه بر این، تکنیک مدیریت منابع تطبیقی برای اطمینان از استفاده بهینه از منابع (تلفن همراه و دستگاه لبه) مورد بررسی قرار خواهند گرفت. اهداف پروژه:

- برای تقلید از سناریوهای واقعی، مجموعه ۵۰ تایی از وظایف دستگاه تلفن همراه، نرم و سخت، توسط الگوریتم uunifast ایجاد کنید.
- پیاده سازی یک الگوریتم زمان‌بندی کارا با الهام از Levy-walk بهینه‌سازی شده برای بهره‌وری انرژی در دستگاه‌های لبه چند هسته‌ای ایجاد کنید. (تعداد تلفن همراه (فردی که تلفن را حمل میکند) = ۵)
- پیاده سازی تکنیک مدیریت منابع تطبیقی برای اطمینان از تخصیص و استفاده بهینه منابع.

محدودیت‌های پیاده سازی:

- محیط مد نظر برای الگوریتم levy walk حدود ۱۰ کیلومتر در ۱۰ کیلومتر به شکل مربعی است.
- ۳۰ دستگاه لبه داریم که هر دستگاه لبه تا شعاع ۱۰۰ متر را پشتیبانی می‌کند و به صورت یکنواخت در محیط پخش شده اند.
- در صورت اجرای وظایف بر روی دستگاه لبه باید سر بار ارسال وظیفه از نظر زمانی و مصرف انرژی مد نظر گرفته شود. (۱) اگر تلفن همراه در محدوده دستگاه لبه باشد می‌توان وظیفه را به دستگاه فرستاد، (۲) در صورت خارج شدن از محدوده یک دستگاه و وارد شدن به محدوده دستگاه لبه دیگر وظیفه به دستگاه لبه هدف ارسال شود. (۳) پیش‌بینی ورود تلفن همراه به محدوده دستگاه لبه تا در صورت نیاز اجرا وظیفه بر روی دستگاه لبه صورت بگیرد (وظیفه منتظر ورود تلفن همراه به محدوده خواهد بود).
- میزان شارژ باتری تلفن همراه بین ۷۰-۹۰ درصد است.
- جهت محاسبه توان مصرفی از فرمول  $P = \alpha C_L V_{dd}^2 f$  استفاده کنید.  $\alpha = 0.1$ ،  $C_L = 1.5-2fF$ ،  $V_{dd} = 0.9-1.1v$ ،  $f = 1.6-2GHz$ .
- هر وظیفه دارای  $\alpha$  مخصوص خود و هر دستگاه لبه و تلفن همراه دارای فرکانس، ولتاژ و خازن موثر مخصوص خود است. (بولت قبل را بررسی کنید). تلفن همراه نباید از دستگاه لبه قوی تر باشد.

خروجی (بهره‌وری: ۰.۳، ۰.۵، ۰.۷):

- نمودار زمان‌بندی پذیری وظایف
- نمودار تعداد تخصیص وظایف بر دستگاه لبه یا تلفن همراه
- نمودار make span، میانگین زمان انتظار و لختی
- نمودار مسیر طی شده توسط افرادی که تلفن همراه دارند و دستگاه‌های لبه موجود در محیط

فاز اول:

- تولید وظیفه، پیاده سازی محیط شبیه سازی الگوریتم Levy-walk و سناریو اول زمان‌بندی وظایف
- ایجاد کانفیگ‌های دستگاه‌های لبه و تلفن همراه
- نمودار مسیر طی شده و دستگاه‌های لبه موجود در محیط

فاز دوم:

- پیاده سازی سناریو دوم و سوم به همراه الگوریتم مدیریت منابع تطبیقی
- تمامی نمودارهای باقی مانده ذکر شده در قسمت خروجی به گزارش اضافه شوند.



یک موشک فضایی شامل چند فضاپرد باید تا تاریخ ۳۰ بهمن ماه ۱۴۰۲ به فضا پرتاب شود و یک گردش چندین ساعته را برای مسافران خود در مدار زمین فراهم کند. مسئله مهم برای تیم طراحی عملکرد درست سیستم اکسیژن فضاپیماست که عملکرد اشتباه آن موجب مرگ مسافران می‌شود. همچنین باید از مسافران در طول این سفر تصویربرداری انجام شود. با توجه به وجود تشعشعات فضایی احتمال رخداد اشکال در این سامانه زیاد است در نتیجه شما باید از روش تحمل پذیری اشکال مبتنی بر رای اکثریت تحت عنوان N-Modular Redundancy (NMR) برای مقابله با این اشکالات استفاده کنید.

شما به عنوان یک مهندس سامانه‌های بی‌درنگ باید یک الگوریتم زمانبندی برای اجرای وظایف بر روی سامانه بحرانی-مختلط این فضاپیما پیاده‌سازی کنید که بصورت زیر باشد.

سامانه بحرانی-مختلط چند هسته‌ای شامل وظایف متناوب LC و HC باشد که وظایف LC دارای یک بدترین زمان اجرا هستند در حالی که وظایف HC دارای دو بدترین زمان اجرا هستند. با توجه به اینکه قابلیت اطمینان سامانه باید حداقل ۰.۹۹۹۹۹۹۹ باشد باید تعداد کپی برای هر یک از وظایف HC را مشخص کنید. سپس وظایف را براساس دو سیاست Worst-Fit Decreasing (WFD) و First-Fit Decreasing (FFD) بر روی هسته‌ها نگاشت کنید. توجه کنید که اولویت با نگاشت وظایف اصلی و کپی‌های آن است. سپس باید وظایف براساس الگوریتم زمانبندی EDF-VD بر روی هسته‌ها زمانبندی شوند. سامانه در حالت نرمال شروع به کار می‌کند و در صورتی که یکی از وظایف HC به اندازه زمان اجرای کوچک خود اجرا شود ولی به پایان نرسد، سامانه وارد حالت overrun می‌شود. توجه نمایید که اجرای درست و به موقع وظایف HC برای سامانه حیاتی است در حالی که نیاز است تا یک حداقل کیفیت خدمات برای سامانه تضمین شود. در نهایت نتایج زیر را باید در گزارش خود ثبت و ارائه کنید:

الف) حالتی که همه هسته‌ها در حالت نرمال کار کنند:

۱. نمودار قابلیت زمانبندی برای سامانه ۸ هسته‌ای با نسبت وظایف HC به LC برابر یک با بهره‌وری ۰.۵.
۲. نمودار کیفیت خدمات برای سامانه ۸ هسته‌ای با نسبت وظایف HC به LC برابر یک با بهره‌وری ۰.۵.
۳. زمانبندی انجام شده برای سامانه ۸ هسته‌ای با نسبت وظایف HC به LC برابر یک با بهره‌وری ۰.۵.

ب) حالتی که ۳۰ درصد هسته‌ها دچار اشکال و ۵۰ درصد هسته‌ها دچار overrun شوند:

۱. نمودار قابلیت زمانبندی برای سامانه ۱۶ هسته‌ای با نسبت وظایف HC به LC برابر یک با بهره‌وری ۰.۷۵.
۲. نمودار کیفیت خدمات برای سامانه ۱۶ هسته‌ای با نسبت وظایف HC به LC برابر یک با بهره‌وری ۰.۷۵.
۳. زمانبندی انجام شده برای سامانه ۱۶ هسته‌ای با نسبت وظایف HC به LC برابر یک با بهره‌وری ۰.۷۵.

در نهایت باید وظایف را با الگوریتم EDF هم زمانبندی کنید و نتایج ذکر شده در بالا را برای این الگوریتم نیز به دست آورید و بررسی کنید که چرا این الگوریتم برای سامانه‌های بحرانی-مختلط مناسب نیست.

فازبندی پروژه:

۱. شما باید در فاز اول به وسیله الگوریتم UUnifast تعدادی وظیفه بحرانی-مختلط تولید کنید و سپس تعداد کپی برای هر وظیفه را مشخص کرده و وظایف را بر روی هسته‌ها نگاشت کنید. (باید نتیجه نگاشت وظایف به هسته‌ها در سامانه ۸ هسته‌ای و ۱۶ هسته‌ای با بهره‌وری‌های ۰.۵ و ۰.۷۵ را در فاز اول ارائه دهید.

۲. در فاز نهایی باید تمام نتایج مورد انتظار گزارش شوند.

پمپ تزریق یک سامانه بحرانی-مختلط است که مقادیر کنترل شده‌ای از مایعات مانند داروها یا مواد مغذی را مستقیماً وارد جریان خون بیمار می‌کند. این سامانه می‌تواند به صورت یک سامانه بحرانی-مختلط دوسطحی که نوعی از سامانه‌های بحرانی-مختلط است در نظر گرفته شود. این نوع سامانه دو سطح بحرانی دارد و شامل وظایف پریودیک LC و HC می‌باشد که وظایف LC دارای یک بدترین زمان اجرا در درجه بحرانی پایین سامانه هستند در حالی که وظایف HC دارای دو بدترین زمان اجرا هستند. سامانه در حالت نرمال شروع به کار می‌کند و در صورتی که یکی از وظایف HC به اندازه زمان اجرای کوچک خود اجرا شود ولی به پایان نرسد سامانه وارد حالت overrun می‌شود. توجه نمایید که اجرای درست و به موقع وظایف HC برای سامانه حیاتی است در حالی که نیاز است تا یک حداقل کیفیت خدمات برای سامانه تضمین شود.

شما به عنوان یک مهندس سامانه‌های بی‌درنگ باید وظایف موردنظر را تحت پروتکل Deadline Floor Inheritance Protocol (DFP) و توسط الگوریتم زمانبندی EDF-VD زمانبندی کنید و در راستای این پیاده‌سازی، منابع باید به گونه‌ای به وظایف اختصاص داده شوند که هم دسترسی تودرتو و هم سریالی به منابع وجود داشته باشد. وظایف در این سامانه باید به صورت مصنوعی و از طریق الگوریتم UUnifast تولید شوند و نتایج باید براساس پارامترهای بهره‌وری، تعداد واحد برای هر منبع، تعداد بخش‌های بحرانی در هر وظیفه، تعداد منابع و نسبت تعداد وظایف HC به LC ثبت و گزارش شوند.

نمودارهای زمانبندپذیری برای بهره‌وری‌های ۰.۳، ۰.۵ و ۰.۷۵:

- تولید ۱۰۰ وظیفه با نسبت تعداد وظایف HC به LC برابر یک، تعداد واحد هر منبع یک مقدار تصادفی بین ۱ تا ۵، تعداد کل منابع برابر با ۱۰ و تعداد بخش‌های بحرانی در هر وظیفه، یک مقدار تصادفی بین ۰ تا ۸.
- تولید ۱۰۰ وظیفه با نسبت تعداد وظایف HC به LC برابر یک، تعداد واحد هر منبع یک مقدار تصادفی بین ۱ تا ۵، تعداد کل منابع برابر با ۱۵ و تعداد بخش‌های بحرانی در هر وظیفه، یک مقدار تصادفی بین ۶ تا ۱۰.

نمودارهای کیفیت خدمات با بهره‌وری‌های ۰.۵ و ۰.۷ و ۰.۹:

- تولید ۱۰۰ وظیفه با نسبت تعداد وظایف HC به LC برابر یک، تعداد واحد هر منبع یک مقدار تصادفی بین ۱ تا ۵، تعداد کل منابع برابر با ۱۰ و تعداد بخش‌های بحرانی در هر وظیفه، یک مقدار تصادفی بین ۰ تا ۸.
- تولید ۱۰۰ وظیفه با نسبت تعداد وظایف HC به LC برابر یک، تعداد واحد هر منبع یک مقدار تصادفی بین ۱ تا ۵، تعداد کل منابع برابر با ۱۵ و تعداد بخش‌های بحرانی در هر وظیفه، یک مقدار تصادفی بین ۶ تا ۱۰.

فازبندی پروژه:

۱. فاز اول: در فاز اول وظایف باید تولید شوند، پیاده‌سازی الگوریتم پیشنهادی برای تولید و نگاشت منابع و تخصیص بخش بحرانی به وظایف و تعیین سطوح قبضگی به عنوان نتایج خروجی گزارش شوند.
۲. فاز دوم: در فاز نهایی، نمودارهای خواسته شده گزارش شوند.

یک سیستم دیجیتال را می‌خواهیم در بالای کوهی قرار دهیم تا اطلاعات جوی و ... که توسط سنسورهایش دریافت می‌کند را برای مرکز ارسال نماید. در این پروژه به زمان‌بندی وظایف در این سیستم بی‌درنگ می‌پردازیم. اما با توجه به محیطی که در آن قرار می‌گیرد نرخ اشکال در آن غیر قابل چشم‌پوشی است، بنابراین باید در این سیستم از تکنیک‌های تحمل‌پذیر اشکال استفاده کنیم. همچنین به دلیل منبع انرژی محدود این سیستم می‌خواهیم انرژی مصرفی آن را نیز کاهش دهیم تا مدت زمان طولانی‌تری انرژی داشته باشد. برای این منظور از دو تکنیک تحمل‌پذیر اشکال مشهور **task replication** و **re-execution** استفاده می‌کنیم.

محدودیت‌های پیاده‌سازی عبارتند از: (۱) برای زمان‌بندی وظایف از الگوریتم EES استفاده کنید که مدیریت انرژی مصرفی را نیز با استفاده از تکنیک محبوب DVFS انجام می‌دهد. (۲) این سیستم شامل مجموعه‌ای از وظایف است که به صورت گراف وظایف (DAG) مدل‌سازی می‌شوند. (۳) سیستم ناهمگن است. (۴) هر وظیفه دارای یک قابلیت اطمینان هدف است که با استفاده از اعمال دو تکنیک تحمل‌پذیر اشکال گفته شده بر روی وظایف، باید این قابلیت اطمینان هدف در بدترین شرایط تضمین شود. (۵) بررسی صحت اجرای هر وظیفه در انتهای اجرای آن صورت می‌گیرد و در صورت درستی در اجرا، وظایف backup آن از سیستم خارج شده و تمام وظایف جلوتر سیستم تا جای ممکن جابجا شده و زودتر اجرا می‌شوند. (۶) مجموعه وظایف ما دارای یک **lower bound** هستند که از طریق الگوریتم معروف HEFT بدست می‌آید. برای شبیه‌سازی موارد زیر را در نظر بگیرید:

- وظایف را به دو صورت Gaussian Elimination و Fast Fourier Transformation تولید کنید (تعداد وظایف را بیش از ۲۲۰ وظیفه در نظر بگیرید).
- قابلیت اطمینان هدف هر عملکرد (گراف) را به صورت تصادفی (با توزیع یکنواخت) و از مجموعه  $\{0.9, 0.99, 0.999, 0.9999, 0.99999\}$  انتخاب کنید.
- زمان اجرای وظایف در بازه  $[10, 100]$  و هزینه ارتباطی را در بازه  $[5, 25]$  به صورت یکنواخت تولید کنید.
- از آنجاکه سیستم ناهمگن است، زمان اجرای وظایف بر روی هر پردازنده و هزینه ارتباطی بین دو وظیفه بر روی هر دو پردازنده باید به صورت تصادفی در بازه گفته شده تولید گردد.
- خروجی‌های مورد نظر: برای هر حالت که در آن مهلت زمانی هر عملکرد (گراف)  $1.1, 1.2$  و  $1.5$  برابر **lower bound** آن است، نمودارهای زیر را تولید کنید (در مجموع ۱۲ نمودار):
- نمودار makespan نهایی سیستم بر حسب نرخ اشکال: برای هر دو تکنیک تحمل‌پذیر اشکال گفته شده این نمودار را برای وظایفی که بر روی ۴ هسته پردازشی اجرا می‌شوند، به ازای نرخ اشکال  $0.0001, 0.0001, 0.0001$  و  $0.0001$  تولید کنید.
- نمودار زمان‌بندی‌پذیری سیستم بر حسب نرخ اشکال: برای هر دو تکنیک تحمل‌پذیر اشکال گفته شده این نمودار را برای وظایفی که بر روی ۴ هسته پردازشی اجرا می‌شوند، به ازای نرخ اشکال  $0.0001, 0.0001, 0.0001$  و  $0.0001$  تولید کنید.
- نمودار انرژی مصرفی سیستم بر حسب تعداد هسته‌های پردازشی: برای هر دو تکنیک تحمل‌پذیر اشکال گفته شده این نمودار را برای وظایفی که بر روی ۲، ۴، ۸ و ۱۶ هسته پردازشی اجرا می‌شوند، به ازای نرخ اشکال  $0.0001$  تولید کنید.
- نمودار انرژی مصرفی نهایی سیستم بر حسب نرخ اشکال: برای هر دو تکنیک تحمل‌پذیر اشکال گفته شده این نمودار را برای وظایفی که بر روی ۴ هسته پردازشی اجرا می‌شوند، به ازای نرخ اشکال  $0.0001, 0.0001, 0.0001$  و  $0.0001$  تولید کنید.

فاز ۱:

- پیاده‌سازی الگوریتم تولید وظایف و تولید یک گراف با آن.
- پیاده‌سازی الگوریتم HEFT و محاسبه **lower bound** هر عملکرد.

فاز ۲:

- پیاده‌سازی تمامی موارد گفته شده، تولید تمام خروجی‌های خواسته شده و تهیه گزارش پایانی.

**توضیحات:** ما در یک محیط شهری هوشمند که با گره‌های محاسباتی مه غنی شده است، به زمان‌بندی وظایف هم‌زمان می‌پردازیم. با استفاده از قدرت یادگیری تقویتی پویا (DRL) و رویکردهای فراابتکاری، این پروژه حول ایجاد یک الگوریتم زمان‌بندی پیشرفته می‌چرخد که وظایف را به طور یکپارچه در چشم‌انداز پویای یک شهر هوشمند هماهنگ می‌کند. لازم به ذکر است برای پیاده‌سازی پروژه باید از ابزار YAFS باید استفاده گردد. اهداف عبارتند از:

- یک الگوریتم زمان‌بندی کار مبتنی بر یادگیری تقویتی پویا متناسب با نیازهای زمان واقعی در یک شهر هوشمند با گره‌های مه ایجاد کنید.
- پیاده‌سازی استراتژی‌های فراابتکاری برای افزایش کارایی الگوریتم زمان‌بندی.
- شبیه‌سازی و ارزیابی عملکرد الگوریتم پیشنهادی با استفاده از معیارهای عملکرد مربوطه.
- تأثیر توزیع گره مه و ویژگی‌های وظیفه را بر اثربخشی الگوریتم زمان‌بندی تجزیه و تحلیل کنید.

بخش بندی پروژه:

#### ۱. الگوریتم زمان‌بندی مبتنی بر یادگیری تقویتی پویا (SCDS) Smart City DRL Scheduler

- یک الگوریتم مبتکرانه مبتنی بر DRL طراحی کنید که یاد می‌گیرد وظایف را به صورت پویا به گره‌های مه اختصاص دهد.
- اجرای SCDS برای بهینه‌سازی مبادله بین استفاده از منابع و تأخیر در حالی که با محدودیت‌های بی‌درنگ سازگار است.
- استراتژی‌های اکتشاف و بهره‌برداری را برای مدیریت عدم قطعیت و یادگیری سیاست‌های زمان‌بندی کارآمد ترکیب کنید.

#### ۲. بهبود توسط الگوریتم فراابتکاری: بهینه‌سازی توزیع گره مه (FNDO)

- یک رویکرد فراابتکاری برای بهینه‌سازی توزیع گره‌های مه در سراسر شهر هوشمند ایجاد کنید.
- از FNDO برای اطمینان از پوشش کافی منابع مه و کاهش تأخیر ارتباط استفاده کنید.
- FNDO را با SCDS ادغام کنید تا کارایی کلی زمان‌بندی کار را افزایش دهید.

خروجی:

- نمودار زمان‌بندی پذیری وظایف بر روی گره‌های مه
- نمودار میانگیر تأخیر و زمان انتظار وظایف
- تاثیر توزیع گره‌های مه بر اجرای وظایف (نگاشت وظایف بر گره‌های مه نیز باید به صورت فایل تکست در گزارش باشد)

فاز اول:

- پیاده‌سازی بخش اول پروژه
- نمایش نمودارهای زمان‌بندی‌پذیری
- نمایش نمودار میانگیر تأخیر و زمان انتظار وظایف

فاز دوم:

- پیاده‌سازی بخش دوم پروژه
- نمایش تمامی نمودارها جهت بررسی تاثیر توزیع گره‌های مه
- نوشتن کامل گزارش

سال ۲۲۰۰ شمسی است و ربات‌های مهاجم زندگی انسان‌ها را به خطر انداخته‌اند. یک ربات نجات‌دهنده نیاز است تا بتواند با شلیک به این ربات‌های مهاجم آن‌ها را نابود کند و سپس از آن عکس بگیرد. در صورت عدم شلیک به موقع توسط ربات مهاجم نابود می‌شود. با توجه به امکان رخداد اشکال بر روی سامانه ربات نجات‌دهنده، برای مقابله با آن باید از روش تحمل‌پذیری اشکال TMR استفاده کنیم. شما به عنوان یک مهندس سامانه‌های بی‌درنگ باید یک الگوریتم زمانبندی برای اجرای وظایف بر روی سامانه بحرانی-مختلط این ربات را ۷۹۸ سال زودتر پیاده‌سازی کنید که بصورت زیر می‌باشد.

سامانه بحرانی-مختلط چندهسته‌ای شامل وظایف پریودیک LC و HC می‌باشد که وظایف LC دارای یک بدترین زمان اجرا هستند در حالی که وظایف HC دارای دو بدترین زمان اجرا هستند. وظایف باید براساس دو سیاست Worst-Fit Decreasing (WFD) و First-Fit Decreasing (FFD) بر روی هسته‌ها نگاشت شوند. توجه کنید که اولویت با نگاشت وظایف اصلی و کی‌های آن است. سامانه از وضعیت نرمال شروع به کار می‌کند. در صورتی که یکی از هسته‌ها وارد حالت overrun شود وظایف LC موجود در آن هسته به هسته‌ی دیگری که زمان خالی دارد منتقل می‌شود هسته‌ای که وظایف LC به آن مهاجرت کرده‌اند وارد یک وضعیت جدید به اسم HOST می‌شوند. این پیش‌بینی باید در هنگام زمان‌بندی لحاظ شود که امکان مهاجرت وظیفه‌ای به آن هسته وجود دارد. سپس باید وظایف براساس الگوریتم زمانبندی EDF-VD بر روی هسته‌ها زمانبندی شوند. توجه نمایید که اجرای درست و به موقع وظایف HC برای سامانه حیاتی است در حالی که نیاز است تا یک حداقل کیفیت خدمات برای سامانه تضمین شود.

در نهایت نتایج زیر را باید در گزارش خود ثبت و ارایه کنید:

الف) حالتی که همه هسته‌ها در حالت نرمال کار کنند:

۱. نمودار قابلیت زمانبندی برای سامانه ۸ هسته‌ای با نسبت وظایف HC به LC برابر یک با بهره‌وری ۰.۵
  ۲. نمودار کیفیت خدمات برای سامانه ۸ هسته‌ای با نسبت وظایف HC به LC برابر یک با بهره‌وری ۰.۵
  ۳. زمانبندی انجام شده برای سامانه ۸ هسته‌ای با نسبت وظایف HC به LC برابر یک با بهره‌وری ۰.۵
- ب) حالتی که ۳۰ درصد هسته‌ها دچار اشکال و ۵۰ درصد هسته‌ها دچار overrun شوند:

۱. نمودار قابلیت زمانبندی برای سامانه ۱۶ هسته با نسبت وظایف HC به LC برابر یک با بهره‌وری ۰.۷۵
۲. نمودار کیفیت خدمات برای سامانه ۱۶ هسته با نسبت وظایف HC به LC برابر یک با بهره‌وری ۰.۷۵
۳. زمانبندی انجام شده برای سامانه ۱۶ هسته با نسبت وظایف HC به LC برابر یک با بهره‌وری ۰.۷۵

در نهایت یک باید وظایف را با الگوریتم EDF زمانبندی کنید و نتایج ذکر شده در بالا را برای این الگوریتم نیز به دست آورید و بررسی کنید که چرا این الگوریتم برای سامانه‌های بحرانی-مختلط مناسب نیست.

فازبندی پروژه:

فاز اول. شما باید در فاز اول به وسیله الگوریتم UUnifast تعدادی وظیفه بحرانی-مختلط تولید کنید و وظایف را بر روی هسته‌ها نگاشت کنید. (باید نتیجه نگاشت وظایف به هسته‌ها در سامانه ۸ تایی و ۱۶ تایی با بهره‌وری ۰.۵ و ۰.۷۵ را در فاز اول ارائه دهید.)

فاز دوم. در فاز نهایی انتظار ارائه تمامی نتایج را از شما داریم.

سامانه بحرانی-مختلط دوسطحی که نوعی از سامانه‌های بحرانی-مختلط است، شامل وظایف با دو درجه بحرانی است که به صورت LC و HC در نظر گرفته می‌شوند. وظایف LC دارای یک زمان اجرا در درجه بحرانی پایین سامانه هستند در حالی که وظایف HC دارای دو بدترین زمان اجرا هستند. سامانه در حالت نرمال شروع به کار می‌کند و در صورتی که یکی از وظایف HC به اندازه زمان اجرای کوچک خود اجرا شود ولی به پایان نرسد سامانه وارد حالت overrun می‌شود. توجه نمایید که اجرای درست و به موقع وظایف HC برای سامانه حیاتی است در حالی که نیاز است تا یک حداقل کیفیت خدمات برای سامانه تضمین شود.

شما به عنوان یک مهندس سامانه‌های بی‌درنگ باید وظایف موردنظر را تحت پروتکل Priority Inheritance Protocol و توسط الگوریتم EDF زمانبندی کنید و در راستای این پیاده‌سازی، منابع باید به گونه‌ای به وظایف اختصاص داده شوند که امکان دسترسی سریالی هر وظیفه به منابع وجود داشته باشد. از آنجایی که سامانه چندهسته‌ای می‌باشد و هر منبع مربوط به یک هسته می‌شود، دسترسی‌های local و global باید مدیریت شوند تا هر وظیفه کمترین میزان انسداد از راه دور را تجربه کند. وظایف در این سامانه باید به صورت مصنوعی و از طریق الگوریتم UUnifast تولید شوند و نتایج باید براساس پارامترهای بهره‌وری، تعداد واحد برای هر منبع، تعداد بخش‌های بحرانی در هر وظیفه، تعداد هسته‌ها و تعداد منابع و نسبت تعداد وظایف HC به LC ثبت و گزارش شوند.

۱- نمودارهای زمان‌بندی‌پذیری و کیفیت خدمات با بهره‌وری‌های ۰.۵ و ۰.۷۵:

- تولید ۱۰۰ وظیفه با نسبت تعداد وظایف HC به LC برابر یک، تعداد واحد هر منبع یک مقدار تصادفی بین ۱ تا ۵، تعداد هسته‌ها برابر ۴، تعداد کل منابع برابر با ۱۰ و تعداد بخش‌های بحرانی در هر وظیفه، یک مقدار تصادفی بین ۰ تا ۱۰.
- تولید ۱۰۰ وظیفه با نسبت تعداد وظایف HC به LC برابر یک، تعداد واحد هر منبع یک مقدار تصادفی بین ۱ تا ۵، تعداد هسته‌ها برابر ۸، تعداد کل منابع برابر با ۱۰ و تعداد بخش‌های بحرانی در هر وظیفه، یک مقدار تصادفی بین ۰ تا ۱۰.

۲- نمودار زمان‌بندی‌پذیری ۵۰۰ وظیفه با نسبت تعداد وظایف HC به LC برابر با یک، تعداد منابع برابر با ۲۰، تعداد هسته برابر با ۴، تعداد بخش‌های بحرانی در هر وظیفه؛ یک مقدار تصادفی بین ۰ تا ۱۰، تعداد واحد هر منبع یک مقدار تصادفی بین ۱ تا ۵ و بهره‌وری ۰.۷۵ برای هر هسته در حالاتی که ۱، ۲ و ۴ هسته تغییر حالت عملیاتی به overrun داشته باشند.

## فازبندی پروژه:

۱. فاز اول: در فاز اول وظایف تولید شده، پیاده‌سازی الگوریتم پیشنهادی برای تولید و نگاشت منابع، تخصیص بخش بحرانی به وظایف و هسته در نظر گرفته شده برای هر وظیفه به عنوان نتایج خروجی گزارش شوند.
۲. فاز دوم: در فاز نهایی، نمودارهای خواسته شده گزارش شوند.

در این پروژه به زمان‌بندی وظایف در یک سیستم ابری می‌پردازیم. در این سیستم عملکردهایی داریم که به صورت پویا وارد سیستم می‌شوند و باید آن‌ها را زمان‌بندی کنیم. در این سیستم رعایت انصاف حائز اهمیت است به گونه‌ای که به هر عملکرد منابع را به طور منصفانه تخصیص دهیم. هدف این پروژه پیاده‌سازی الگوریتم‌هایی برای زمان‌بندی چنین سیستمی است. سه الگوریتم  $FDWS$ ،  $MinMax$  و  $RANK\_HYBD$  برای این منظور ارائه شده‌اند. این سه الگوریتم مشابه را پیاده‌سازی کرده و آن‌ها را با یکدیگر مقایسه نمایید.

محدودیت‌های پیاده‌سازی:

- در این سیستم هر عملکرد شامل مجموعه‌ای از وظایف است که به صورت گراف وظایف (DAG) مدل‌سازی می‌شوند.
- عملکردها به صورت پویا وارد سیستم می‌شوند.
- سیستم ناهمگن است.
- $Unfairness$  معیاری برای مقایسه الگوریتم‌های گفته شده است. این مقدار را برای الگوریتم‌های گفته شده بدست آورید از روابط زیر بدست می‌آید. در این روابط  $a$  یک عملکرد است،  $M_{own}(a)$  مقدار  $makespan$  این عملکرد هنگامی که به صورت مجزا زمان‌بندی می‌شود و  $M_{multi}(a)$   $makespan$  این عملکرد زمانی که با بقیه عملکردها زمان‌بندی شده است، می‌باشد.

$$Unfairness = \sum |slowdown(a) - avg\_slowdown|$$

$$Slowdown(a) = \frac{M_{own}(a)}{M_{multi}(a)}$$

شبیه‌سازی:

- وظایف هر عملکرد را به دو صورت  $Gaussian\ Elimination$  و  $Fast\ Fourier\ Transformation$  تولید کنید.
- میانگین زمان اجرای وظایف در بازه  $[10, 100]$  و هزینه ارتباطی را در بازه  $[5, 25]$  به صورت یکنواخت تولید کنید.
- زمان اجرای واقعی وظایف را توزیع نرمال با انحراف معیار ۵ در نظر بگیرید. بدترین زمان اجرا را ۲۵ واحد بیشتر از میانگین لحاظ کرده و بر اساس بدترین زمان اجرا وظایف را زمان‌بندی کنید. اگر وظیفه‌ای زودتر از میانگین زمان اجرای خود به اتمام رسید، تمام وظایف جلوتر سیستم تا جای ممکن جابجا شده و زودتر اجرا می‌شوند.
- از آنجاکه سیستم ناهمگن است، زمان اجرای وظایف بر روی هر پردازنده و هزینه ارتباطی بین دو وظیفه بر روی هر دو پردازنده باید به صورت تصادفی در بازه گفته شده تولید گردد.
- زمان رسیدن عملکردها را به صورت تصادفی و در بازه  $[0, 1000]$  انتخاب کنید.

خروجی‌های مورد نظر:

برای ۳ حالت که در آن تعداد وظایف هر عملکرد بیش از ۲۲۰، ۵۰۰ و ۱۰۰۰ است، نمودارهای زیر را تولید کنید (در مجموع ۱۲ نمودار):

- نمودار  $makespan$  نهایی سیستم بر حسب تعداد عملکردها: برای هر سه الگوریتم گفته شده این نمودار را برای ۴، ۸، ۱۶ و ۳۲ عملکرد که در یک سیستم با ۸ هسته پردازشی اجرا می‌شوند، تولید کنید.
- نمودار  $makespan$  نهایی سیستم بر حسب تعداد هسته‌های پردازشی: برای هر سه الگوریتم گفته شده این نمودار را برای ۴، ۸، ۱۶ و ۳۲ عملکرد که در یک سیستم با ۸ هسته پردازشی اجرا می‌شوند، تولید کنید.
- نمودار  $Unfairness$  نهایی سیستم بر حسب تعداد عملکردها: برای هر سه الگوریتم گفته شده این نمودار را برای ۴، ۸، ۱۶ و ۳۲ عملکرد که در یک سیستم با ۸ هسته پردازشی اجرا می‌شوند، تولید کنید.

- نمودار Unfairness نهایی سیستم بر حسب تعداد هسته‌های پردازشی: برای هر سه الگوریتم گفته شده این نمودار را برای ۴، ۸، ۱۶ و ۳۲ عملکرد که در یک سیستم با ۸ هسته پردازشی اجرا می‌شوند، تولید کنید.

فاز ۱:

- پیاده‌سازی الگوریتم تولید وظایف و تولید ۸ عملکرد با آن.
- پیاده‌سازی الگوریتم MinMax و زمان‌بندی ۸ عملکرد تولید شده با استفاده از الگوریتم MinMax (بدون جابجایی وظایف در صورت اجرای زودتر از انتظار).

فاز ۲:

- پیاده‌سازی تمامی موارد گفته شده، تولید تمام خروجی‌های خواسته شده و تهیه گزارش پایانی.



## توضیحات پروژه:

### توسعه الگوریتم زمان‌بندی پیشرفته:

- ایجاد یک الگوریتم زمان‌بندی پیشرفته و نوآورانه بر مبنای تکنیک‌های بهینه‌سازی مانند الگوریتم‌های ژنتیک کوانتومی یا الگوریتم جک اسپارو به منظور بهینه‌سازی مصرف انرژی و زمان‌بندی وظایف.
- این الگوریتم باید قابلیت بلادرنگی را نیز پشتیبانی کند و بتواند وظایف را به سرعت به دستگاه‌های لبه یا تلفن‌های همراه منتقل کند.

### پیاده‌سازی تکنیک مدیریت منابع تطبیقی:

- ایجاد تکنیک‌های مدیریت منابع تطبیقی (تخصیص دینامیک منابع و مدیریت منابع مبتنی بر اولویت) که به طور دینامیک منابع مثل CPU، حافظه، شبکه را به وظایف اختصاص دهد تا مصرف انرژی و زمان‌بندی بهینه‌تری را فراهم کند.
- این تکنیک‌ها باید بتوانند به تغییرات در محیط وارد شده (مانند ورود یا خروج تلفن‌های همراه به محدوده دستگاه‌های لبه) پاسخ دهند.

### نمودارهای خروجی:

- نمودار میزان بهره‌وری انرژی
- نمودار make span (زمان انجام وظایف)
- نمودار میانگین زمان انتظار و نمودار مدت زمان ارسال و دریافت داده توسط تلفن همراه
- نمودار مسیر طی شده توسط تلفن‌های همراه
- نمودار دستگاه‌های لبه نیز جهت نمایش مسیرهای انتقال وظایف

### محدودیت‌های پیاده‌سازی:

- ایجاد یک محیط شبیه‌سازی با ابعاد و مشخصات (محیط مربعی با شعاع ۱۰ کیلومتر و ۲۰ دستگاه لبه) جهت اجرای الگوریتم‌ها و تست‌ها.
- استفاده از ۳۰ الی ۵۰ دستگاه موبایل ناهمگن از نظر منابع محاسباتی با توزیع نرمال جهت حضور در محیط شبیه‌سازی
- استفاده از مدل‌های مصرف انرژی برای تلفن‌های همراه و دستگاه‌های لبه بر اساس فرمول معروف.
- تکنیک‌ها و الگوریتم‌ها باید به تغییرات در محیط و ورود و خروج تلفن‌های همراه پاسخ دهند.
- دستگاه‌های لبه ناهمگن هستند و میتوانند بسته به شرایط شعاع ۱۰۰ الی ۲۰۰ متری را ساپورت کنند و میتوانند از ۳-۷ تعداد دستگاه موبایل را پشتیبانی کنند.

### فاز اول:

- ایجاد محیط شبیه‌سازی با ابعاد و ویژگی‌های مشخص
- پیاده‌سازی الگوریتم‌های زمان‌بندی و تخصیص وظایف
- تولید وظایف با تکنیک unifast
- اجرای سناریو اول زمان‌بندی وظایف و ایجاد نمودارهای مربوطه.

### فاز دوم:

- پیاده‌سازی سناریوهای دوم و سوم به همراه تکنیک‌های مدیریت منابع تطبیقی.
- ایجاد نمودارهای باقی‌مانده طبق توضیحات داده شده.
- ارزیابی عملکرد نهایی الگوریتم‌ها و تکنیک‌ها با استفاده از مقادیر خروجی و نمودارهای ایجاد شده.

**توضیحات:** این پروژه، به زمانبندی وظایف با هدف کاهش Makespan در سیستم‌های چند هسته‌ای همگن Non-preemptive می‌پردازد. در سیستم مورد نظر می‌توان با اختصاص دادن تعداد بیشتری هسته به یک وظیفه، زمان اجرای آن را کاهش داد؛ اما پس از اختصاص یک هسته به یک وظیفه تا زمان پایان آن وظیفه نمی‌توان هسته را از آن گرفت.

### الگوریتم Cooperative

در این الگوریتم به ترتیب وظایف را اجرا و به هر کدام از آنها هسته‌ها را اختصاص می‌دهیم. این الگوریتم به عنوان Baseline در نظر گرفته شود.

### الگوریتم Best

این الگوریتم تمامی حالات زمانبندی را بررسی و حالت با کمینه Makespan را انتخاب می‌کند. در صورت یکسان بودن Makespan چند حالت، انرژی مصرفی و Peak Power معیارهای بعدی خواهند بود. هدف از پیاده‌سازی این الگوریتم داشتن حد بالایی برای Makespan است.

### الگوریتم Profile

در این الگوریتم قصد داریم تابع زیر را بیشینه کنیم.

$$\sum_{i=1}^N S(i, j)^{\frac{1}{N}}$$

که در آن  $S(i, j)$  برابر تسریع وظیفه  $i$ ام با  $j$  واحد پردازشی است.

در لحظه صفر همه هسته‌ها با توجه به این الگوریتم بین همه وظایف تقسیم می‌شوند. بعد از پایین هر وظیفه هسته‌های آزاد شده مجدداً با این الگوریتم بین وظایف باقی مانده توزیع می‌شوند.

**ورودی‌ها:** پروفایل شامل زمان اجرا، توان، و انرژی مصرفی هر وظیفه به ازای تعداد هسته‌ها (حداکثر ۶ هسته)

### خروجی‌ها:

- نمودار زمانبندی الگوریتم‌ها
- نمودار مقایسه تسریع الگوریتم‌ها بر حسب تعداد وظیفه
- نمودار مقایسه انرژی مصرفی الگوریتم‌ها بر حسب تعداد وظیفه
- نمودار مقایسه سربار زمانی الگوریتم‌ها بر حسب تعداد وظیفه

### فاز اول:

- تولید مجموعه وظایف با ۲ تا ۶ وظیفه با توجه به پروفایل‌ها
- نمودار زمانبندی الگوریتم‌ها

### فاز دوم:

- رسم سایر نمودارها

