



دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

سیستم های نهفته و بی درنگ

(بهار ۱۴۰۲)

تمرین عملی دوم

استاد درس:

دکتر حامد فربه

مقدمه :

امروزه سیستم های نهفته توان پایین در زمینه های مختلف استفاده می شوند. به عنوان مثال در خودروهای خودران ربات های زنده یاب و ... اشاره کرد. برای رفع توان پایین بودن سامانه اغلب ما با محدودیت استفاده از پردازنده هایی با هسته های زیاد مواجه هستیم. از جهت دیگر کارهای سامانه دریافت مقادیر و انجام تسک ها باید بی درنگ انجام شود. و کار های حیاتی باید اولویت بیشتری داشته باشند. به همین دلیل جهت برآورده کردن نیازهای بالا به سیستم عامل های سبک و با حداقل مصرف منابع نیاز داریم.

به همین دلیل سیستم عامل های بی درنگ برای بستر های مختلف سخت افزاری به وجود آمدند به عنوان مثال:

- Free RTOS
- ROS (Robot Operating system)
- Contiki

اما اکثر آن های دارای محدودیت هایی هستند برای مثال سیستم عامل محبوب FreeRTOS بر روی پردازنده های AMD توانایی اجرا به صورت صحیح را ندارد.

در این تمرین سعی داریم که با استفاده از نرم افزار جهانی Python یک سیستم عامل کوچکی را طراحی کنیم که در ادامه به آن می پردازیم.

سیستم عامل ما باید توانایی مدیریت تسک ها و زمان بندی آن ها را داشته باشد و در صورت از دست رفتن یک تسک به نوعی آن را هندل کند.

سیستم عاملی که توسعه می دهیم دارای موجودیت های زیر است .

Task:

هر تسک ۴ از نوع aperiodic , sporadic, periodic interrupt می تواند باشد .

علاوه بر این تسک ها دارای ۳ حالت هستند که وضعیت آن ها را نشان میدهد .

Running,Ready,Suspended

و همچنین دارای ویژگی های دیگری مانند priority , id نیز می باشند.

: Task Set

مجموعه از تسک ها هستند که دارای انواع تسک می باشد .

برای تسک ست ها ما ویژگی های زیر را داریم :

بهره وری (utility)

مدت زمان اجرا

feasibility که این ویژگی پس از اجرا شدن task set مشخص می شود.

:Scheduler

برای زمان بندی تسک ست ها ما نیاز به یک الگوریتم زمان بندی داریم .

در این تمرین شما نیاز است که الگوریتم های زمان بندی , RM , EDF, DM را پیاده سازی کنید .

: Printer

این موجودیت تنها وظیفه این را دارد که خروجی زمانبندی انتخاب شده با تسک ست داده شده را نمایش دهد.

که اطلاعاتی شامل نمودار تسک ها (مانند simso) , زمان ها , feasible بودن , و بهره وری سیستم را نمایش دهد.

در نهایت با استفاده از تمامی این ها ما باید RTOS خود را پیاده سازی کنیم .

به صورتی که ورودی تسک های ما از فایل csv که همراه تمرین قرار داده شده است خوانده شده و نتایج مشخص می شود.

**** امتیازی : پیاده سازی الگوریتم EDF به هر دو صورت**

preemptive & non preemptive

• لینک کد های مربوطه در [اینجا](#) قرار دارد.

سوالاتی که باید به آن پاسخ داده شود :

در هر تسک ست داده شده کدام الگوریتم مناسب تر است . (دلیل آن را توضیح دهید)

نکاتی در رابطه با نحوه ارسال تمرین :

- ❖ پاسخ به تمرین ها باید به صورت فردی انجام شود.
- ❖ هرگونه شباهت با یکدیگر و موارد مشابه در اینترنت موجب کسر نمره ۱۰۰٪ ای میشود.
- ❖ این تمرین نیازمند گزارش نیز میباشد که در قالب یک فایل PDF باید تحویل دهید.
- ❖ نکاتی که در گزارش باید باشد:
 - نحوه پیاده سازی به صورت خلاصه
 - نحوه کارکرد توابع و کلاس ها
 - اسکرین شات از پاسخ های خود
- ❖ پاسخ خود را در یک فایل PDF بصورت خوانا در سامانه کورسز آپلود کنید.
- ❖ فرمت نامگذاری تمرین باید مانند HW2_9931099.pdf باشد.
- ❖ در صورت هرگونه سوال یا مشکل با ایمیل درس و یا آیدی @lostago در تلگرام در تماس باشید.
- ❖ ددلاین این تمرین ۱۵ اردیبهشت ۱۴۰۲ ساعت ۲۳:۵۹ است و امکان تمدید آن به هیچ وجه وجود ندارد.