گزارش نتایج کارگاه

Asghar abbasnezhad |operating system concepts | 8,may,2017

# الگوریتم first come first serve(FCFS)

در این الگوریتم ، اولین پردازه ای که درخواست می دهد اولین پردازی ای خواهد بود که به خدمت گرفته می شود پس در این صورت ما یک Arrival time که مشخص کننده ای زمان رسیدن پردازه خواهد بود و همچنین یک burst time خواهیم داشت که نشان دهنده ای زمان انفجار پردازه خواهد بود.

در این الگوریتم زمان انتظار برای پردازه های با زمان انفجار کم که در پشت پردازه های با زمان انفجار بالایی هست خیلی زیاد می شود.

بنابراین چیزی که در این الگوریتم قابل توجه است میانگین زمان انتظار است.که بهترین شیوه ای که من در این الگوریتم به نظرم رسید که آن را بهبود دهد این بود که پردازه های را که در زمان یکسانی می رسند بر اساس زمان انفجار انها به صورت صعودی مرتب کنیم در غیر این صورت براساس زمان رسیدن پردازه ها انها را به صورت صعودی مرتب کرده و انها را وارد پردازنده کنیم تا پردازش شوند که در این صورت هم بهره وری پردازنده کاهش می یابد.

همیچنین در این الگوریتم وقتی پردازه ای وارد پردازنده میشود تا زمانی که زمان انفجارپردازه پایان نیابد ازپردازنده خارج نمیشود واز نوع nonpreemptive است .

و با توجه به این که پردازه ها با چه ترتیب زمانی به پردازنده می رسند میانگین زمان انتظار انها متفاوت خواهد بود .

## الگوریتم round-robin

این الگوریتم براساس برش زمانی تعریف میشود که هر برش زمانی به صورت چرخشی هر بار در اختیار یک پردازه قرار میگیرد تا زمانی که همه ای پردازه ها در پردازنده ،پردازش شده باشند.

برای الگوریتم FCFS دیدم که با چینش های مختلف پردازه ها برای ورود به پردازنده(در صورتی که زمان رسیدن انها به پردازنده برای همه ای پردازه ها یکسان باشد.)، مقدار میانگین زمان انتظار تغییر میکرد.

در این الگوریتم زمان گردش را خواهیم داشت که مجموع دوره های زمانی است که پردازنده در انتظار ورود به حافظه انتظار سپری صف اماده ، اجرا در صف پردازنده وانجام i\o سپری مینماید.که در فایل ضمیه با اسم round-robin.c با متغییر Turnaround محاسبه می شود.

البته قابل ذکر است که در پیاده سازی الگوریتم های مذکور مقدار دهی تصادفی هم به صورت کامنت تعبیه شده که برای شبیه سازی اینکه زمان انفجار و زمان رسیدن پردازه ها توسط پردازنده مشخص می شود.

## نتیجه گیری

کارایی الگوریتمROUND-ROBIN وابستگی زیادی به کمیت زمانی دارد.اگر کمیت زمانی را خیلی بزرگ بگیریم کارایی ان شبیه FCFS خواهد بود(TIME QUANTUM =max(P1 ,P2,P3,P4…..)) و اگر کمیت زمانی را بسیار کوچک در نظر بگیریم به تعداد زیادی برگردان متن منجر میشود .

همچنین الگوریتم ROUND-ROBIN پس گرفتنی است و اگر تعداد پردازه ها N تا باشد و Qکمیت زمانی باشد در ان صورت در هر دوره چرخش هر پردازه حداکثر Q (N-1) انتظار میکشد تا پردازنده را در اختیار بگیرد.ولی در الگوریتم FCFS امکان دارد که حتی به یکی از پردازه ها اصلا پردازنده، در اختیار آن قرار نگیرد .

زمان گردش( turnaround time )هم بستگی به اندازه کمیت زمانی دارد.به طور کلی موقعی میانگین زمان گردش بهبود می یابد که بیشتر پردازه ها انفجاری بعدی پردازنده ای خود را تنها با یک کمیت زمانی منفرد به اتمام برسانند.

میانگین زمان انتظار برای این الگوریتم برای پردازه های مشخص با زمان انتظار مشخصی مقدار ثابتی دارد.یعنی وقتی پردازه های با Arrival time یکسان به پردازنده میرسند میانگین زمان انتظار انها با هر جایگشتی از پردازه ها یکسان است ولی در الگوریتم FCFS این وضعیت بر قرار نیست لزوما.

بنابراین در الگوریتم ROUND-ROBIN تقریبا همه ای معیار های زمان بندی به کمیت زمانی بستگی دارد.

این فایل و همچنین دو فایل ضمیمه round-robin.c و FCFS.c در github بارگذاری شده اند و از ادرس <https://github.com/asqarabb/os-workshop>

قابل دست رسی خواهد بود.