توضیح موارد : FCFS , Non Preemptive SJF , Priority With Peemptiom :

به دلیل حجم بسیار بالای سه کد و کامل بودن کامنت گزاری ها لذا تصمیم بنده این است که استراکت کلی کد ها خدمت شما به عنوان توضیح عرض کنم وهمچنین از بعضی موارد مشابه در کد ها هم صرف نظر نمایم . برای پیاده سازی هر سه مورد کد هم ورودی پردازه ها از فایل متنی گرفته شده و هم از طریق ورودی کامند لاین (بخش های کار با فایل در کد با کامنتگذاری مشخص شده اند) و همچنین تمرکز گسترش کد بیشتر بر اساس ورودی های کامند لاینی است . اما کد ها به طور کلی سه بخش دارند :

بخش اول : بخش کار با فایل و یا گرفتن ورود از کاربر که در هرسه مشابه است در در کاربافایل فایل متنی حاوی اعداد توسط کد موجود باز شده خط به خط از هم جدا شده و داخل یک آرایه دو بعدی ریخته شده اند و در بخش دریافت ورودی هم عملکردئ مشخص است و تمام ورودی ها در کامند لاین گرفته می شوند و در 4 آرایه یک بعدی ذخیره میشوند .

بخش دوم : در این بخش سه کد تفاوت های اساسی دارند در FCFS همانطور که اولین کد با کمترین ایندکس اولین ورودی است پس از چک اریوال تایم که با تابع time اجرا میشود و(و برعدد 100 باقی مانده گرفته میشود تا مدل مفهومی ایجاد کند) از صف خارج می شود . اما در کد SJF ما باید به غیر از چک کردن اریوال تایم باید بیاییم و کوتاه ترین burst time را پیدا کنیم آنرا اجرا و از درون لیست حذف کنیم (که کار یافتن کوتاه ترین زمان توسط تابع می نیمم یاب که روی آرایه CT عمل می کند انجام می شود) اما در Priority چون علاوه بر چک کردن اریوال تایم به روش های قبلی باید این بار هر بار پردازه با بیشترین اولویت را یافته و از صف حذف کنیم که انجام این عمل توسط یک تابع ماکسی مم یاب که روی آرایه priority اجرا میشود پیاده سازی میشود اما در این بخش نکته مهم دیگر غیر انحصاری بودن است که توسط یک حلقه تو در تو که یکی تعداد دفعات دسترسی به پردازنده و دیگری نوبت ها را بررسی می کند انجام میشود و تایم هر نوبت 2 در نظر گرفته شد واز طریق شروط موجود اگر تابعی زمان burst time اش صفر شود دیگر از صف حذف میشود .

بخش سوم : اما بخش سوم که تقریبا در تمام کدها یکی است انجام محاسبات زمانی است در اینجا برای پیاده سازی از تابع ()process_time در بخش های مختلف تمام کد ها(بخش آغاز و میانه (یعنی شروع الگوریتم اصلی) و پایان هر کد) استفاده شده تمام زمانها را به صورت دقیق داشته باشیم . اما برا utiliztion چون پردازنده دائم استفاده میشود 100% و برای طول کل زمان process_time اولی منهای آخری وتعداد پردازه هم تعداد ورودی یا تعداد پردازه های موجود در متن است .through put برابر تقسیم تعداد پردازه ها به کل زمان است که مقدار هردو را از قبل داریم . برای waiting time و response time هم همانطور که میدانیم زمان اجرای هر پردازش زمان صبر پردازش های دیگر به استثنا آخرین زمان اجرا که در آن کسی صبر نمی کند پس برای FCFS و SJF جمع تمام زمانهای اجرا به جز آخری و برای priorityچون غیر انحصاری است و زمان اجرای آخر همان 2 (حداکثر) است پس همان دو را از مجموع زمان های اجرا کم می کنیم ور در آخر برای میانگین تقسیم بر تعداد پردازه ها می کنیم البته همگی باید در آخر جمع اریوال تایم ها از آنها کم شود .turn around هم فاصله شروع ازپایان آخرین اجرا است پس طول تمام اجراها میشود و آخرین اجرا هم کاسته نمیشود سپس تقسیم بر تعداد پردازه ها میکنیم تا میانگین بدست آید در اینجا هم کم کردن مجمع اریوال تایم ها مهم است البته برای اینکه تقریب دقیق تری داشته باشیم همه یه جمع هایی که از مجموع زمان پردازش ها داریم را باید در نصف تعداد پردازه ها ضرب کنیم تا میانگین انتظار همه بدست آید .برای هر سه تای(turn around ,waiting time ,priority)این کار لازم است. نکته مهم در هنگام ورودی دادن چه در فایل چه کامند لاین اگر ورودی ها و به ویژه اریوال تایم با نسبت درست ودقیقی وارد نشوند محاسبات انتهایی بسیار غیر عقلانی میشوند مثلا اگر اریوال تایم ها را به اشتباه بزرگ بدهیم میزان میانگین زمان انتظار منفی میشود لذا وروردی دقیق محاسبات را نیز دقیق می کند.