بسم الله الرحمن الرحیم

شبیه‌ساز حافظه‌ی مجازی

Virtual Memory Simulator

# مقدمه

یکی از تکنولوژی‌های مدیریت حافظه در سیستم‌عامل‌ها، حافظه‌ی مجازی نام دارد. در این روش، ممکن است تمام داده‌های پردازه‌های درحال‌اجرا در پردازنده، درون حافظه‌ی اصلی موجود نباشند و تنها در هنگام نیاز، از دیسک به حافظه‌ی اصلی منتقل شوند. این تکنولوژی امکان اجرای پردازه‌های بیشتری را برای پردازنده فراهم می‌کند که تا حدی از بیکاری پردازنده جلوگیری می‌شود.

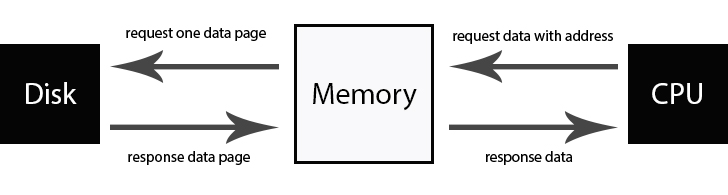
مدیریت حافظه به روش حافظه‌ی مجازی می‌تواند گونه‌های مختلفی داشته باشد. یکی از عوامل موثر در تعدد گونه‌ها، مبحث پارتیشن‌بندی حافظه است که بدین منظور می‌توان از روش‌های صفحه‌بندی(paging)، قطعه‌بندی(segmentation)، یا ترکیب این دو روش استفاده نمود. مبحث دیگر این حوزه الگوریتم‌های مختلف مبادله‌ی داده‌ها میان دیسک و حافظه‌ی اصلی است.

در این پروژه، برای پارتیشن‌بندی حافظه، از روش صفحه‌بندی و برای مبادله داده‌ها از الگوریتم FIFO استفاده‌شده‌است. برای پیاده‌سازی این پروژه، اقدام به ایجاد محیطی به صورت یک برنامه کاربردی تک‌صفحه‌ای تحت وب (SPA) شده است.

# شرح کلی

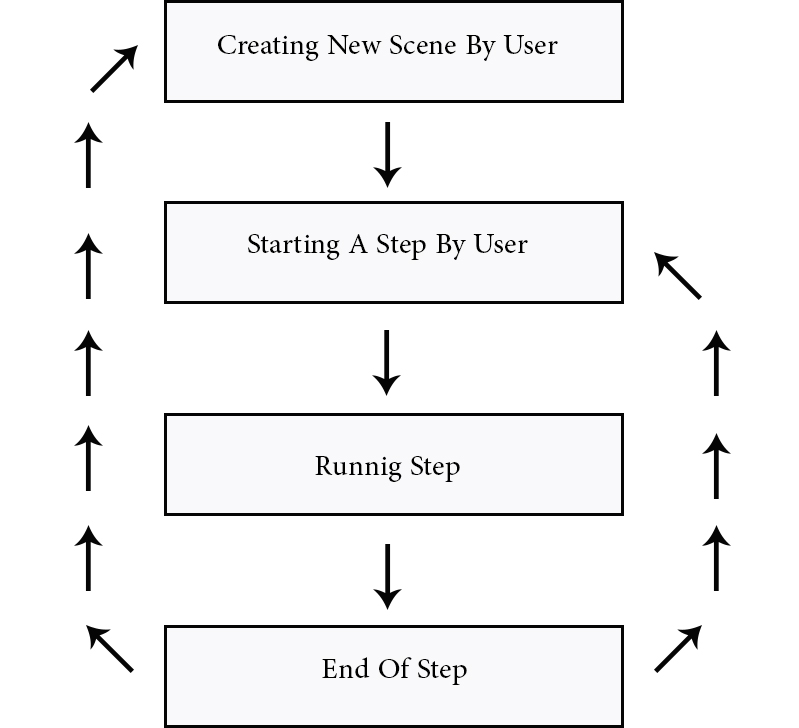
در این پروژه سه مؤلفه موردنیاز است؛ دیسکی که تمام داده‌های پردازه‌ها را در اختیار دارد، پردازنده‌ای که پردازه‌ها را اجرا می‌کند، و حافظه اصلی که قرار است مدیریت شود. موضوع پروژه مدیریت حافظه است، ازاین‌رو تنها جزئیات حافظه اهمیت دارد. بنابراین دیسک و پردازنده به صورت جعبه‌سیاه درنظر‌گرفته شده‌اند که تنها نیازهای پروژه برای پیاده‌سازی مدیریت حافظه را برآورده می‌کنند.. درمورد دیسک تنها دو موضع اهمیت دارد؛ اولا تمام داده‌ها را در اختیار داشته باشد و ثانیا درهنگام درخواست یک مجموعه‌ی بخصوص از داده‌ها، آن مجموعه را در اختیار درخواست‌کننده قراردهد. درمورد پردازنده، تنها یک موضوع اهمیت دارد؛ درخواست داده‌ با یک آدرس معتبر.

ارتباطات این سه مؤلفه مانند شکل 1 خواهد بود. پردازنده می‌تواند از حافظه داده با آدرس مشخص را درخواست نماید و حافظه به آن پاسخ می‌دهد. حافظه نیز می‌تواند یک مجموعه‌داده (یک صفحه از داده‌های یک پردازه) را از دیسک درخواست نماید و دیسک به آن پاسخ می‌دهد.



شکل 1 (ارتباطات سه مؤلفه‌ی اصلی)

پروسه سیستم نیز بدین گونه خواهد بود که کاربر ابتدا پیکربندی محیط را انجام می‌دهد. درواقع کاربر در ابتدای امر یک سناریو و صحنه یا سیستم فرضی جدید ایجاد می‌کند. سپس با هر بار درخواست کاربر سیستم یک گام اجرایی را طی می‌کند که این گام با درخواست داده بخصوصی به‌وسیله‌ی پردازنده از حافظه آغاز و با تحویل داده به‌وسیله‌ی حافظه به پردازنده خاتمه میابد. پس از هر گام نیز کاربر میتواند دوباره عملیات پیکربندی را انجام دهد (شکل 2).



شکل 2 (پروسه‌ی سیستم)

# شرح با جزئیات

در این قسمت به بررسی دقیق‌تری درباره‌ی پروسه‌ی سیستم می‌پردازیم.

## پیکربندی/ صحنه‌ی جدید/ سناریوی جدید

در مرحله، کاربر سه مشخصه اصلی سناریو را مشخص می‌کند که عبارتنداز:

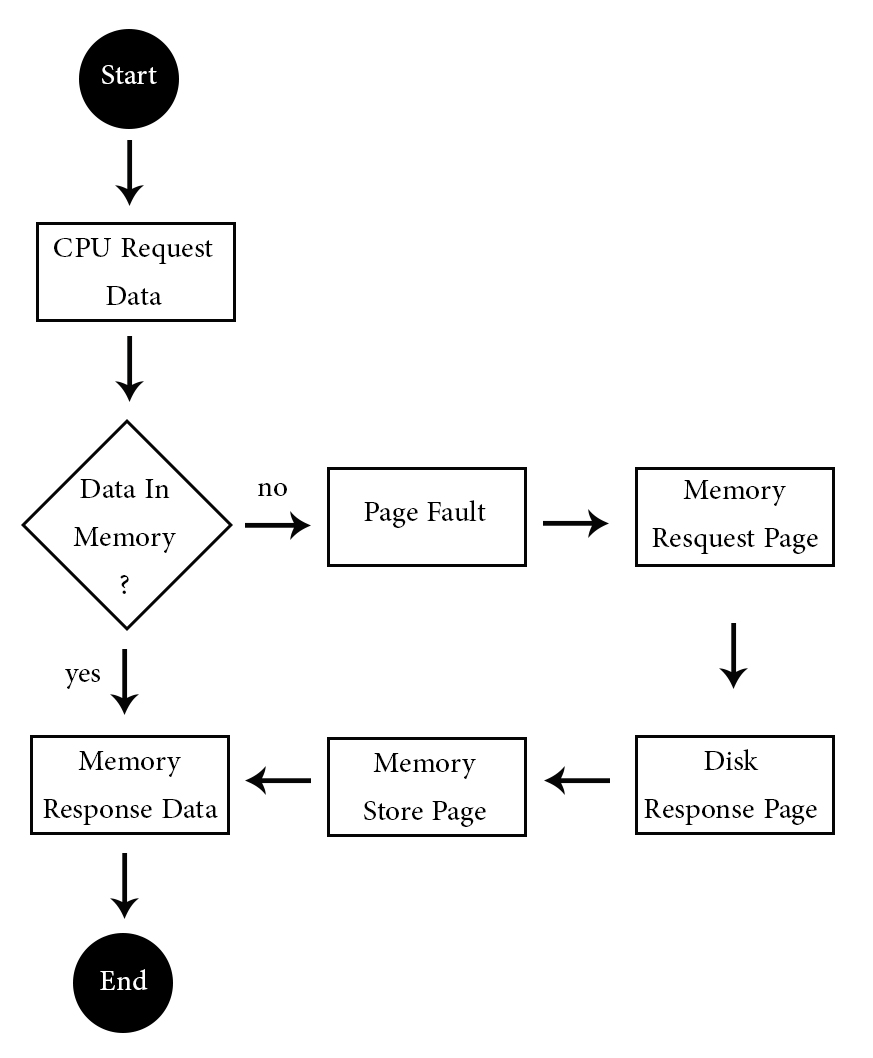
1. اندازه هر فریم/صفحه در واحد بایت
2. اندازه حافظه در واحد فریم
3. مجموعه‌ی پردازه‌های سیستم که خود شامل سه قسمت شماره، اندازه داده‌های آن در واحد صفحه، و رنگ است. شماره به‌طور خودکار و به ترتیب با افزودن پردازه مقداردهی می‌شود (از صفر) و رنگ نیز به طور تصادفی برای هر پردازه انتخاب می‌شود.

پس از مقداردهی موارد فوق و انتخاب گزینه‌ی ایجاد، سیستم پیکربندی می‌شود. حافظه براساس اندازه‌‌ی فریم و اندازه‌ی حافظه ایجاد می‌شود و مقدار هر بایت آن در ابتدا عدد صفر قرار می‌گیرد. پردازه‌ها نیز با توجه به اطلاعات وارد شده برای پردازه، درون دیسک ایجاد می‌شوند و مقدار هر بایت داده آن‌ها به‌صورت تصادفی انتخاب می‌شود (برای ملموس‌کردن انتقال داده‌ها در هنگام بصری‌سازی). پردازنده نیز با داشتن اطلاعات مربوط به پردازه‌ها می‌تواند آدرس معتبری از حافظه درخواست کند.

پس از اتمام مراحل ذکرشده‌ی فوق، سیستم آماده‌ی اجرا است.

## گام‌ اجرایی

کاربر با انتخاب گزینه‌ی شروع، یک گام اجرایی را آغاز می‌کند. با شروع یک گام، ابتدا پردازنده یک آدرس معتبر تصادفی ایجاد و از حافظه داده‌ی موجود در آن آدرس را درخواست می‌کند. در مرحله‌ی دوم مدیر حافظه وجود داده‌ی مورد نظر را در حافظه بررسی می‌کند. اگر داده‌ی موردنظر در حافظه موجود باشد، داده، به پردازنده تحویل داده می‌شود و سپس گام اجرایی پایان می‌یابد. اگر داده‌ی مورد نظر در حافظه موجود نباشد، یک خطای صفحه رخ می‌دهد. در هنگام رخداد خطای صفحه، مدیر حافظه صفحه‌ای که داده‌ی موردنظر را داراست، از دیسک دخواست می‌کند. سپس دیسک، صفحه موردنظر را به حافظه تحویل می‌دهد و مرحله‌ی ذخیره‌سازی صفحه موردنظر در حافظه آغاز می‌شود. اگر حافظ فریم خالی داشته باشد، صفحه در اولین فریم خالی ذخیره می‌شود. در غیر این‌صورت، یکی از صفحات موجود با استفاده از الگوریتم FIFO حذف و صفحه‌ی جدید جایگزین آن می‌شود. سپس داده‌ی موردنظر درون صفحه به پردازنده تحویل داده می‌شود و گام اجرایی پایان می‌یابد (شکل 3).



شکل 3 (مراحل یک گام اجرایی)

## مدیر حافظه

بررسی وجود داده‌ی موردنظر برحسب آدرس مشخص‌شده، و تبدیل آدرس منطقی به آدرس فیزیکی و واکشی داده توسط آدرس فیزیکی، فرآیندی است که مدیر حافظه، با استفاده از یک جدول صفحه انجام می‌دهد. جدول صفحه موجود شامل سه ستون می‌باشد؛ شماره‌ی فریم حافظه، شماره‌ی پردازه‌ای که صفحه‌ی آن درون فریم قراردارد، و شماره‌ی صفحه‌ی پردازه. تعداد رکوردهای جدول می‌تواند به تعداد فریم‌های حافظه باشد.

الگوریتم FIFO که برای جایگزینی صفحات استفاده می‌شود بدین شرح است که مدیرحافظه همواره یک ساختارداده‌ی صف نگهداری می‌نماید. هنگام ذخیره‌ی یک صفحه درون یک فریم حافظه، شماره‌ی فریم به انتهای صف اضافه می‌شود. اگر فریم خالی برای ذخیره‌ی صفحه وجود نداشته باشد، مدیر حافظه ابتدای صف را بررسی می‌کند و صفحه جدید را با صفحه‌ای که شماره‌ی فریم آن در ابتدای صف قراردارد جایگزین می‌کند و شماره فریم موردنظر را از ابتدای صف خارج کرده و به انتهای صف اضافه می‌کند.

## نکات پایانی

اولین نکته این است که، کاربر هر زمان که بخواهد می‌تواند سناریوی جدید ایجاد کند و سیستم را از نوع پیکربندی نماید. دومین نکته این است که، کاربر می‌تواند بی‌نهایت گام اجرایی را درون یک سناریو آغاز نماید.