به نام خدا

دانشجو امیر تیموری

تمرین جلسه دوم

حافظه Stack and Heap

به طور کلی حافظه Heap بخشی از حافظه کامپیوتر شما است که به صورت خودکار برای شما مدیریت نمی‌شود، و به صورت محکم و مطمئن توسط پردازنده مرکزی مدیریت نمی‌شود. آن بیشتر به عنوان یک ناحیه شناور بسیار بزرگی از حافظه است

بر خلاف حافظه اِستک (Stack) حافظه هیپ محدودیتی در اندازه متغیرها ندارد (جدا از محدودیت آشکار فیزیکی در کامپیوتر شما). حافظه هیپ در خواندن کمی کُند تر از نوشتن نسبت به حافظه اِستک است، زیرا جهت دسترسی به آن‌ها در حافظه هیپ باید از اشاره گر استفاده شود. بر خلاف حافظه اِستک، متغیرهایی که در حافظه هیپ ساخته می‌شوند توسط هر تابعی در هر بخشی از برنامه شما در دسترس بوده و اساسا متغیرهای تعریف شده در هیپ در دامنه سراسری قرار دارند.

حافظه اِستک(Stack)

ناحیهٔ اِستک (Stack) شامل برنامه اِستک، با ساختار **LIFO**کوتاه شده عبارت **Last In First Out** (آخرین ورودی از همه زودتر خارج می‌شود) به طور رایج در بالاترین بخش از حافظه قرار می‌گیرد. یک (اشاره گر پشته) در بالاترین قسمت اِستک قرار می‌گیرد. زمانی که تابعی فراخوانی می‌شود این تابع به همراه تمامی متغیرهای محلی خودش در داخل حافظه اِستک قرار می‌گیرد و با فراخوانی یک تابع جدید تابع جاری بر روی تابع قبلی قرار می‌گیرد و کار به همین صورت درباره دیگر توابع ادامه پیدا می‌کند.

مزیت استفاده از حافظه اِستک در ذخیره متغیرها است، چرا که حافظه به صورت خودکار برای شما مدیریت می‌شود. شما نیازی برای اختصاص دادن حافظه به صورت دستی ندارید، یا نیازی به آزاد سازی حافظه ندارید. به طور کلی دلیل آن نیز این است که حافظه اِستک به اندازه کافی توسط پردازنده مرکزی بهینه و سازماندهی می‌شود. بنابراین خواندن و نوشتن در حافظه اِستک بسیار سریع است.

کلید درک حافظه اِستک در این است که زمانی که تابع خارج می‌شود، تمامی متغیرهای موجود در آن همراه با آن خارج و به پایان زندگی خود میرسند. بنابراین متغیرهای موجود در حافظه اِستک به طور طبیعی به صورت محلی هستند. این مرتبط با مفهوم دامنه متغیرها است که قبلا از آن یاد شده است، یا همان متغیرهای محلی در مقابل متغیرهای سراسری.

یک اشکال رایج در برنامه نویسی C تلاش برای دسترسی به یک متغیر که در حافظه اِستک برای یک تابع درونی ساخته شده است می‌باشد. یعنی از یک مکان در برنامه شما به خارج از تابع (یعنی زمانی که آن تابع خارج شده باشد) رجوع می‌کند.

یکی دیگر از ویژگی‌های حافظه اِستک که بهتر است به یاد داشته باشید این است که، محدودیت اندازه (نسبت به نوع سیستم عامل متفاوت) است. این مورد در حافظه هیپ صدق نمی‌کند.

خلاصه ای از حافظه اِستک(Stack)

- حافظه اِستک متناسب با ورود و خروج توابع و متغیرهای درونی آن‌ها افزایش و کاهش می‌یابد
- نیازی برای مدیریت دستی حافظه برای شما وجود ندارد، حافظه به طور خودکار برای متغیرها اختصاص و در زمان نیاز به صورت خودکر آزاد می‌شود
- در اِستک اندازه محدود است
- متغیرهای اِستک تنها در زمان اجرای تابع ساخته می‌شوند
- مزایا و معایب حافظه اِستک و هیپ

حافظه اِستک(Stack)

- دسترسی بسیار سریع به متغیرها

- نیازی برای باز پس گیری حافظه اختصاص یافته شده ندارید
- فضا در زمان مورد نیاز به اندازه کافی توسط پردازنده مرکزی مدیریت می‌شود، حافظه ای نشت نخواهد کرد
- متغیر ها فقط محلی هستند
- محدودیت در حافظه اِستک بسته به نوع سیستم عامل متفاوت است
- متغیر ها نمی‌توانند تغییر اندازه دهند

حافظه هیپ(Heap)

حافظه Heap در قست space-user حافظه مجازی قرار دارد و به صورت دستی توسط برنامه نویس مدیریت می شود. Heap مربوط به زمان اجرا (runtime) است و فضای اشغال شده در heap با اتمام کار تابع آزاد نمی شوند و تا زمانی که Collector Garbage این فضا را آزاد کند یا توسط برنامه نویس داده ها از حافظه heap پاک نشوند در این فضا باقی می ماند. اندازه حافظه heapمتغیر است به همین دلیل به آن memory dynamic گفته می شود .در این نوع از حافظه برای ذخیره مقادیر ابتدا محاسبه ای توسط سیستم عامل صورت می گیرد تا اولین فضای حافظه ای که اندازه آن متناسب با اندازه ای که مورد نیاز ماست را پیدا کند، در صورت وجود این میزان از حافظه درخواستی آن را به صورت رزرو شده درمی آورد تا بقیه برنامه ها به این فضا دسترسی نداشته باشند، سپس آدرس ابتدای این فضای محاسبه شده به صورت یک اشاره گر (pointer) در اختیارمان قرار می دهد (یا به اصالح allocating).متغیر ها به صورت پیش فرض در این حافظه قرار نمی گیرند و اگر قصد ذخیره متغیر ها در این حافظه را داشته باشیم باید به صورت دستی این اقدام انجام شود. متغیر هایی که در heapذخیره می شوند به طور خودکار حذف نمی شوند و باید توسط برنامه نویس و به صورت دستی حذف شوند. به طور کلی مدیریت حافظه heapبه صورت دستی توسط برنامه نویس انجام می شود. آرایه های داینامیک در heap ذخیره می شوند .در صورتی که داده های ما از تعدا block های پشت سر هم در حافظه بیشتر باشد یا در صورت تغییر حجم داده ها در زمان های مختلف)تغییر سایز داده ها امکان پذیر است(، سیستم عامل داده ها را به صورت تکه تکه در block های حافظه ذخیره خواهد کرد .به دلیل محاسبات برای یافتن آدرس شروع حافظه و در اختیار گرفتن pointer حافظه heap نسبت به stack کندتر است. همچنین اگر داده ها به صورت پشت سر هم در block های حافظه قرار نگرفته باشند)این احتمال بسیار زیاد است(موجب کندی در بازیابی اصالعات خواهد شد.

- متغیرها به صورت سراسری قابل دسترس هستند
- محدودیتی در اندازه حافظه وجود ندارد
- تضمینی برای حافظه مصرفی وجود ندارد، ممکن است حافظه در زمان‌های خاص از برنامه نشت کرده و حافظه اختصاص یافته شده برای استفاده در عملیات دیگر آزاد نخواهد شد
- شما باید حافظه را مدیریت کنید، شما باید مسئولیت آزاد سازی حافظه های اختصاص یافته شده به متغیرها را بر عهده بگیرید