به نام خدا

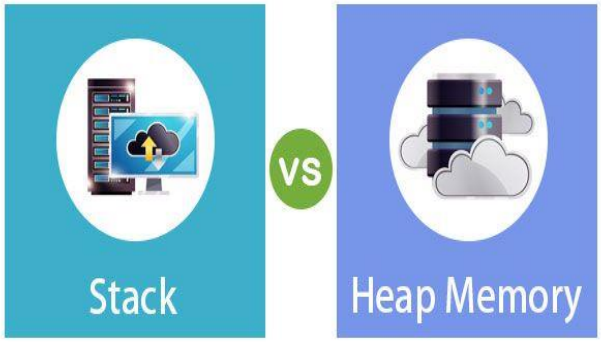
میلاد کلاته

درس برنامه نویسی سمت سرور

استاد آقای میثاق یاریان

شماره دانشجویی: 01221033720030

حافظه Heap و Stack



تعریف یک متغیر، ساختinstance ، توابع، ایجاد یک thread و ... در برنامه نیاز به ذخیره سازی در حافظه دارند و همه آن ها فضایی از حافظه را اشغال می کنند. بنابراین لازم است تا یک برنامه نویس درک درستی از حافظه ای که در اختیار می گیرد داشته باشد. در این مقاله دو نوع از حافظه هایی که در زمان توسعه در اختیار برنامه نویس قرار می گیرد را بررسی می کنیم.

حافظه Stack؟

در بخش user-space حافظه قرار دارد و به صورت خودکار توسط CPU مدیریت می شود. متغیرهای غیر استاتیک، پارامتر های ارسالی به توابع و آدرس های مربوط به return توابع در این حافظه ذخیره می شوند. اندازه حافظه stack ثابت است به همین دلیل به آن static memory گفته می شود.

در این حافظه اطلاعات پشت سر هم و به ترتیب قرار می گیرند به این صورت که آخرین داده ذخیره شده در بالای stack قرار می گیرد و به اصطلاح push می شود، حال اگر قصد برداشتن اطلاعات یا به اصطلاح pop کردن اطلاعات را داشته باشیم آخرین اطلاعات وارد شده در stack را در اختیار داریم. به این الگوریتم LIFO(Last In First Out) می گویند. مثال پر کاربرد در توضیح stack خشاب اسلحه (آخرین گلوله ای که در خشاب قرار داده می شود اولین گلوله ای است که شلیک می شود) و یا بشقاب های روی هم چیده شده (آخرین بشقابی که روی سایر بشقاب ها قرار داده می شود اولین بشقابی است که برداشته می شود) است.

از آنجا که در حافظه stack نیازی به پیدا کردن فضای خالی در حافظه نیست و محل قرارگیری اطلاعات مشخص است (بالای حافظه) بنابراین این حافظه سریع تر از حافظه heap است.

پارامتر ها و اطلاعات مربوط به توابع برای اجرا و کنترل آن ها در این حافظه ذخیره می شوند. تابعی که در بالای stack قرار دارد تابعی است که در حال اجراست و بعد از اتمام کار تابع یا بروز خطا در اجرای تابع، حافظه اختصاص داده شده به تابع از stack حذف می شود و حافظه اشغال شده آزاد می شود. زمانی که یک thread تعریف می شود در stack قرار می گیرد.

خطایی که ممکن است در اثر استفاده نادرست از حافظه stack رخ دهد stack overflow است. از جمله دلایل stack overflow یا سر ریز می توان به استفاده از متغیرهای محلی حجیم که منجر به کاهش فضای آزاد در stack و تخریب یا corrupt شدن بخشی از memory اشاره کرد.

حافظه Heap

حافظه Heap در قست user-space 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 Garbage Collector این فضا را آزاد کند یا توسط برنامه نویس داده ها از حافظه heap پاک نشوند در این فضا باقی می ماند. اندازه حافظه heap متغیر است به همین دلیل به آن dynamic memory 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 اصلاح.(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 heap ذخیره می شوند به طور

خودکار حذف نمی شوند و باید توسط برنامه نویس و به صورت دستی حذف شوند. به طور کلی مدیریت حافظه heap به 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 stackکندتر 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 اصلاعات خواهد شد.

وقتی که نمونه‌ای از یک کلاس ایجاد می‌کنیم این مقدار در Heap ذخیره می‌شود و وقتی که کار آن به پایان می رسد garbage collector حافظه را آزاد می کند و اگر موفق به این کار نشود، برنامه نویس باید به صورت دستی حافظه heap را آزاد کند، در غیر این صورت Memory leak اتفاق می افتد که به معنی in use نگه داشتن فضای حافظه برای اشیایی است که دیگر از آن ها در برنامه استفاده نمی شود و garbage collector قادر به آزاد سازی فضایی که آن ها اشغال کرده اند نیست.

به طور کلی Value Type ها (primitive type) فضای زیادی اشغال نمی کنند و در stack ذخیره می شوند. برای دسترسی به متغیر هایValue Type ، مقدار آن به صورت مستقیم از حافظه stackخوانده می شود، مثلا زمانی که متغیری تعریف می کنیم آن متغیر به همراه مقدار آن در stackقرار می گیرد.

برای دسترسی به متغیرهایReference Type ، ابتدا با مراجعه Stack و دریافت آدرس متغیر در Heapبه شئ مربوط به متغیر دسترسی خواهیم داشت Reference Type .ها در حافظه heap نگهداری می شوند. زمانی که یک شی از کلاس ایجاد می کنیم ابتدا متغیری که شی به آن assign شده است با مقدار null در حافظه stack قرار می گیرد، سپس شی در heap ذخیره شده و پس از ذخیره سازی در heap آدرس شی در stack جایگزین null می شود.
همچنین reference type ها به dynamic memory و value type ها به static

memoryنیاز دارند. در صورت نیاز بهdynamic memory ، باید امکان دسترسی به heap فراهم باشد و اگر نیازمند static memory باشیم، stackمحل ذخیره سازی خواهد بود.

تفاوت Value Type و Reference Type :

وقتی یک متغیر از نوع value type تعریف می کنیم، قسمتی از حافظه stack گرفته می شود و مقدار آن متغیر نیز در stack ذخیره می شود. برای مثال وقتی یک متغیر از نوع int 32 تعریف می کنیم، چهار بایت از حافظه stack گرفته می شود و مقدار آن نیز در stack نگهداری می شود. اما وقتی یک reference type تعریف می کنیم، یک قسمت از حافظه stack گرفته می شود که داخل آن یک آدرس نگهداری می شود و این آدرس به قسمتی از managed heap اشاره می کند. مقدار آن نیز در managed heap نگهداری می شود.

در نتیجهvalue type ها در stack تعریف و نگهداری می شوند و برایreference type ها، آدرس در stack و مقدار در managed heap نگهداری می شود.

یکی از امکاناتی که دات نت در اختیار ما قرار می دهد مدیریت حافظه از طریق garbage collectorاست. به این صورت که در managed heap متغیرها را بررسی می کند و اگر به ازای آنها در stack آدرسی وجود نداشته باشد حافظه را آزاد می کند.

انواعی که در سی شارپ به عنوان value type در نظر گرفته می شوند عبارتند از:

انواع عددی:

sbyte, short, int, long

انواع اعشاری:

float, double, decimal

انواع داده های دو وضعیتی که true و false می باشد و بعد از آن کاراکترها هستند.

در مقابل reference type های پیش فرض سی شارپ که اولین آن ها object می باشد. تمام کلاس هایی که در سی شارپ تعریف می کنیم از یک کلاس به نام object ارث بری می کنند.

یکی دیگر از reference type های پرکاربرد، رشته ها می باشند که به کمک کلمه کلیدی string تعریف می شوند. اصلاحا به آن ها immutable گفته می شود. به متغیرهایی که مقدار آن ها می تواند تغییر کنند متغیرهای mutable گفته می شود. اما در مقابل متغیرهایی هستند که اگر بخواهیم مقدار آن ها را عوض کنیم باید ماهیت آن ها را عوض کنیم. به این نوع متغیرها immutable گفته می شود. متغیرها از نوع رشته نیز یکی از آنها می باشد.

برای مثال وقتی یک متغیر string تعریف می کنیم، آدرس آن در stack و مقدار آن در heap نگهداری می شود. اگر من بخواهم مقدار متغیر تعریف شده را تغییر دهم، یک خانه جدید در heap ایجاد می شود و مقدار جدید را نگهداری می کند و آدرس قبل در stack به مقدار جدید اشاره می کند. بنابراین یک متغیر جدید ایجاد می شود و مقدار جدید در آن قرار می گیرد.