

به نام خدا

نام و نام خانوادگی: امیرحسین نظری یامچلو

شماره دانشجویی: 01221033720039

نام استاد: آقای میثاق یاریان

نام درس: برنامه نویسی سمت سرور

تمرین شماره 2

موضوع: حافظه ی heap و stack

و

value type و reference type

## حافظه stack

در بخش user-space حافظه قرار دارد و به صورت خودکار توسط CPU مدیریت می شود. متغیرهای غیر استاتیک، پارامترهای 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 (Last In First Out)(LIFO) 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.

خطایی که ممکن است در اثر استفاده نادرست از حافظه stack رخ دهد stack overflow است. از جمله دلایل stack overflow یا سرریز 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 قسمت 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 اصطلاح allocating).

متغیرها به صورت پیش فرض در این حافظه قرار نمی گیرند و اگر قصد ذخیره متغیرها در این حافظه را داشته باشیم باید به صورت دستی این اقدام انجام شود. متغیرهایی که در heap ذخیره می شوند به طور خودکار حذف نمی شوند و باید توسط برنامه نویس و به صورت دستی حذف شوند. ب

ه 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 تعداد 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 اطلاعات خواهد شد.

همچنین reference type ها به dynamic memory و value type ها به static memory نیاز دارند. در صورت نیاز به dynamic memory، باید امکان دسترسی به heap فراهم باشد و اگر نیازمند static memory باشیم، stack محل ذخیره سازی خواهد بود.

در زبان برنامه نویسی سی شارپ (#C)، متغیرهایی که از نوع مرجع (reference type) استفاده می کنند، شامل کلاس ها، آرایه ها، رشته ها، اشیاء و تمامی کلاس هایی هستند که از کلاس های دیگر به عنوان پایه ی خود استفاده می کنند.

در این نوع متغیرها، مقدار متغیر به صورت آدرس حافظه ای به محتوای آن اشاره می کند و تغییر در مقدار آن متغیر، تغییری در محتوای آن آدرس حافظه ای ایجاد می کند.

اما متغیرهایی که از نوع مقدار (value type) استفاده می کنند، شامل اعداد صحیح (integers)، اعداد اعشاری (floats)، کاراکترها (characters)، بولین ها (booleans) و ساختارهای ساده ای هستند که تنها شامل یک مقدار هستند.

در این نوع متغیرها، مقدار متغیر به صورت مستقیم در حافظه ی مربوط به آن ذخیره می شود و تغییر در مقدار آن متغیر، تغییری در مقدار آن حافظه ی مستقیم ایجاد می کند.

در زبان‌های برنامه‌نویسی، متغیرها می‌توانند در دو منطقه اصلی حافظه ذخیره شوند: `heap` و `stack`.

متغیرهایی که در زمان کامپایل مقدار آن‌ها مشخص است و حجم آن‌ها ثابت است، در `stack` ذخیره می‌شوند. این شامل متغیرهای اعداد صحیح، اعشاری، رشته‌ای، بولی و غیره است.

متغیرهایی که در زمان اجرا ایجاد می‌شوند و حجم آن‌ها ممکن است در طول اجرا تغییر کند، در `heap` ذخیره می‌شوند. این شامل متغیرهایی است که با استفاده از عبارت `new` در زبان‌هایی مانند `C`، `++Java` و `C#` ایجاد می‌شوند. همچنین، آرایه‌هایی که در زمان اجرا ایجاد می‌شوند و حجم آن‌ها مشخص نیست، نیز در `heap` ذخیره می‌شوند.

در کل، استفاده از `heap` و `stack` بستگی به نوع متغیر و نحوه تعریف آن دارد.