

به نام خدا

امیر سجاد خوش صاحبی

نام درس: طراحی و واسط برنامه نویسی شی گرا

نام استاد: جناب آقای میثاق یاریان

موضوع : حافظه heap stack

stack یک بخش خاصی از حافظه هست که برای ذخیره متغیرهای محلی هر تابع از جمله تابع main و محل برگشت از تابع استفاده میشه. معمولا طراحی stack به صورت First In Last Out هست یا همون FILO. خوبی stack این هست که توسط CPU مدیریت میشه و کاربر نیاز نداره که به صورت دستی حافظه رو Allocate کنه و بعد از پایان استفاده اون رو deallocate کنه. در CPU مکانیزی وجود داره که به صورت اتوماتیک متغیرهای محلی آدرس دهی میشن و در نهایت هم deallocate میشن. از اونجا که این کار توسط CPU انجام میشه خیلی سرعت بالایی داره و تقریبا هیچ گونه Memory Leakage رخ نمیده (به این معنا که حافظه ایی به اشتباه پر فرض بشه). همونطور که گفتیم خوبی استفاده از stack این هست که حافظه برای شما به صورت اتوماتیک مدیریت میشه. نکته مهم در فهم کار stack دانستن این نکته هست که وقتی از یک تابع خارج میشویم تمام متغیرهای محلی این تابع از داخل stack خارج میشن یا همون pop میشن و کلا از بین میروند. نکته مهم دیگه این هست که حافظه stack محدود هست. این در مقایسه با حافظه heap هست. به صورت کلی stack :

- ۱- حافظه stack به صورت اتوماتیک با توجه به سبایز متغیرهای محلی , pop یا push میشه.
- ۲- هیچ نیازی به مدیریت حافظه از طرف کاربر نیست و متغیرها به صورت اتوماتیک allocate و deallocate میشن.
- ۳- stack -داری محدودیت اندازه هست.
- ۴- متغیرهای توابع که داخل stack ذخیره شدن تا وقتی که cpu داخل اون تابع هست دسترسی هستند و وقتی که cpu از اون تابع بیرون میره تمام متغیرهای اون تابع پاک میشن

حالا حافظه heap چی هست؟؟؟

حافظه heap قسمتی از حافظه هست که به صورت اتوماتیک مدیریت نمیشه بنابراین کاربر باید بعد از اختصاص دادن حافظه heap به یک متغیر حواش باشه که اون متغیر رو از حافظه پاک کنه. اگر این اتفاق به هر دلیلی نیفته میگن memory leakage اتفاق افتاده و به این معنا هست که ما یک قسمت از حافظه رو از دست دادیم چون cpu فکر میکنه متغیر داخل اون قسمت وجود داده و نمیتونه جای اون متغیر چیزه دیگه ایی بگذاره. اختصاص دادن حافظه heap به متغیر توسط دستور alloc() انجام میشه که یک تابع داخلی C هست. وظیفه پاک کردن متغیر از حافظه heap به عهده کاربر هست و باید اون متغیر رو با دستور free() از حافظه پاک کنه. برخلاف stack حافظه heap ظرفیت بیشتری داره و همچنین محدودیت روی ساز متغیرها هم وجود نداره. از اونجا که مدیریت heap توسط کاربر انجام میشه سرعت کار کردن باهاش کمتر از stack خواهد بود. از طرف دیگه متغیرهای ذخیره شده در heap به صورت global هستند و همه توابع میتونن به اون متغیرها دسترسی

داشته باشند. به طور خلاصه, در حافظه heap :

- ۱- متغیرها به صورت global قابل دسترسی هستند.
- ۲- میشه سرعت دسترسی کمتر از stack خواهد بود.
- ۳- احتمال memory leakage وجود داره.
- ۴- متغیرها به صورت داینامیک هستند و میتونن با استفاده از دستور realloc() تغییر سبایز بدهند.

حالا این سوال پیش میاد که کی از heap استفاده کنیم و کی از stack؟؟؟

اگر قرار هست به مقدار بزرگی از دیتا برای مدت زمان زیاد دسترسی داشته باشید, باید از heap استفاده کنید. اگر قرار هست که متغیرها رو به صورت محلی در توابع استفاده کنید باید از stack استفاده کنید.