

به نام خدا

موضوع : پشته & هیپ

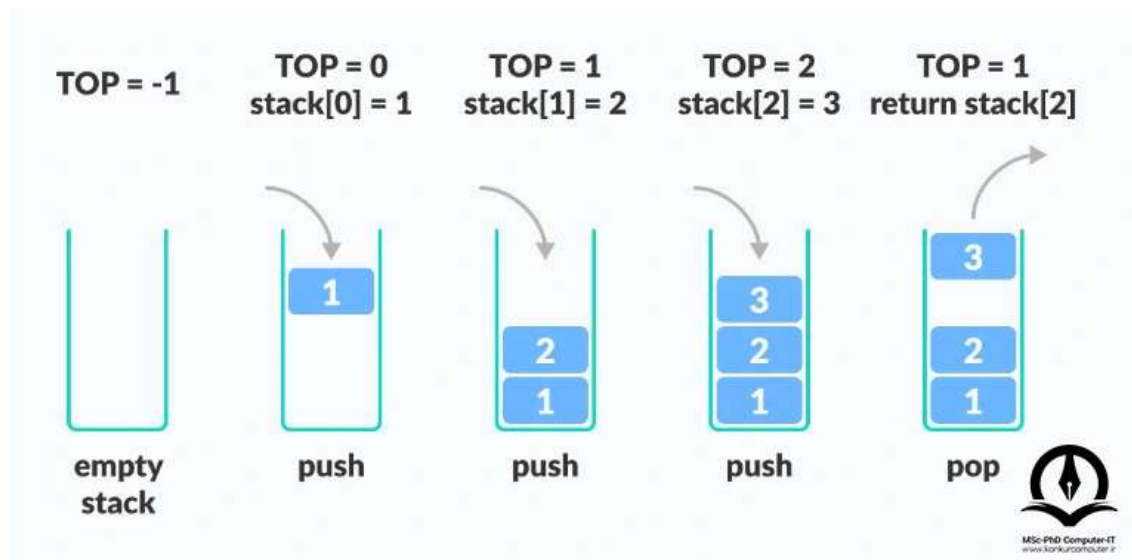
دانشجو : عرفان معمارپور

استاد : میثاق یاریان

درس : طراحی صفحات وب پیشرفته

پشته و درخت هیپ دوتا از پرکاربردترین ساختمان داده ها در زبان برنامه نویسی میباشند.

پشته قسمتی از حافظه میباشد که برنامه متغیرهای محلی و تاریخچه فراخوانی توابع را در آن نگه داری میکند . این ساختمان داده از الگوی LIFO استفاده میکند و آخرین المنتی که به داخل پشته وارد میشود ، به عنوان اولین المنت از پشته خارج میشود . وقتی یک متد فراخوانی میشود پارامترها و متغیرهای آن داخل پشته تخصیص داده میشوند . پشته نسبت به درخت هیپ از سرعت بالایی برخوردار میباشد



دو عملیات در پشته وجود دارد Push و Pop

در عملیات Push یک عنصر جدید داخل پشته قرار میگیرد و در عملیات Pop آخرین عنصر اضافه شده از پشته خارج میشود

در پشته برای اضافه کردن عنصر یا برای برداشتن آن باید همیشه جایگاه بالاترین عنصر را بدانیم.

برای این منظور یک متغیر به نام TOP وجود دارد که مقدارش همیشه برابر با جایگاه بالاترین عنصر است. در ابتدا که پشته، خالی است و هیچ عنصری در آن وجود ندارد، مقدار TOP برابر ۱ است. با هر بار اضافه کردن عنصر ، مقدار TOP یکی اضافه میشود یعنی با Push کردن اولین عنصر، مقدار TOP برابر ۰ خواهد شد. با Push کردن عنصر بعدی، مقدار TOP برابر ۱ خواهد شد و ... با هر بار برداشتن عنصر از پشته (Pop)، مقدار TOP یکی کاهش میابد.

در پشته دوخطا ممکن است رخ دهد:

خطای سرریز: هنگامی که برای پشته ظرفیت تعیین کرده باشیم، در صورتیکه پشته پر باشد و بخواهیم یک عنصر دیگر اضافه کنیم خطای سرریز رخ می دهد.

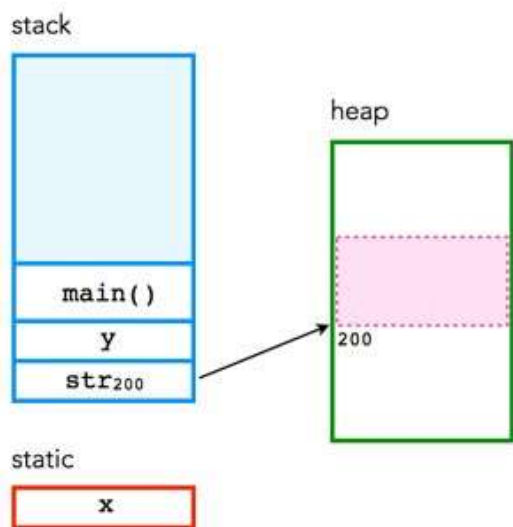
خطای پاریز: در صورتیکه پشته خالی باشد و ما بخواهیم عملیات Pop را انجام بدهیم خطای پاریز رخ خواهد داد.

یکی دیگر از ساختمان داده های جذاب دنیای کامپیوتر درخت هیپ هست. هیپ یک درخت باینری و کامل میباشد ، در این درخت دودویی هر نود حداکثر دو فرزند میتواند داشته باشد ، هر نود حداقل یک فرزند دارد که از خود کوچکتر است و حداقل یک فرزند دارد که از خود بزرگتر میباشد . ساختمان داده های هیپی که بزرگترین مقدار در ریشه قرار دارد max هیپ نامیده میشوند و به همین ترتیب min هیپ نوعی هیپ است که در آن، هر مقدار گره کوچکتر از مقادیر فرزندانش است و گره حاوی کوچکترین مقدار میباشد.

حافظه هیپ در قسمت user-space حافظه مجازی قرار دارد و به صورت دستی توسط برنامه نویس مدیریت می شود .هیپ مربوط به زمان اجرا است و فضای اشغال شده در هیپ 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 هیپ پاک نشوند در این فضا باقی می ماند. اندازه حافظه هیپ 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 در اختیارمان قرار می دهد . 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 هیپ 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 هیپ به صورت دستی توسط برنامه نویس انجام می شود. آرایه های داینامیک در هیپ ذخیره می شوند.

Value Type ها فضای زیادی اشغال نمی کنند و در پشته ذخیره می شوند. برای دسترسی به متغیر های Value Type ، مقدار آن به صورت مستقیم از حافظه پشته خوانده می شود، مثلاً زمانی که متغیری تعریف می کنیم آن متغیر به همراه مقدار آن در پشته قرار می گیرد. برای دسترسی به متغیر های Reference Type ، ابتدا با مراجعه به پشته و دریافت آدرس متغیر در هیپ به شیء مربوط به متغیر دسترسی خواهیم داشت. Reference Type ها در حافظه هیپ نگهداری می شوند. زمانی که یک شیء از کلاس ایجاد می کنیم ابتدا متغیری که شیء به آن اختصاص داده شده است با مقدار null در حافظه پشته قرار می گیرد، سپس شیء در هیپ ذخیره شده و پس از ذخیره سازی در هیپ آدرس شیء در پشته جایگزین null می شود.



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 هیپ فراهم باشد و اگر نیازمند static memory باشیم، پشته محل ذخیره سازی خواهد بود.