

به نام خدا

محمدجواد قمری

دانشجوی رشته فناوری اطلاعات

طراحی و برنامه نویسی شیء گرا

استاد آقای میثاق یاریان

شماره دانشجویی 01221033720028

تمرین

میزان حافظه ذخیره سازی برای variable type دیتابیس

INT

بایت 1: TINYINT

بایت 2: SMALLINT

بایت 3: MEDIUMINT

بایت 4: INT

بایت 8: BIGINT

DOUBLE و FLOAT

بایت 4: FLOAT

بایت 8: DOUBLE

VARCHAR و CHAR

بایت (حداکثر 255 بایت) n: CHAR(n)

بایت (حداکثر 65535 بایت) n+1: VARCHAR(n)

TEXT

TINYTEXT: حداکثر 255 بایت

TEXT: حداکثر 65535 بایت (64 کیلوبایت)

MEDIUMTEXT: حداکثر 16,777,215 بایت (16 مگابایت)

LONGTEXT: حداکثر 4,294,967,295 بایت (4 گیگابایت)

BLOB

TINYBLOB: حداکثر 255 بایت

BLOB: حداکثر 65535 بایت (64 کیلوبایت)

MEDIUMBLOB: حداکثر 16,777,215 بایت (16 مگابایت)

LOB: حداکثر 4,294,967,295 بایت (4 گیگابایت)

DATE و TIME

DATE: 3 بایت

DATETIME: 8 بایت

TIMESTAMP: 4 بایت

TIME: 3 بایت

YEAR: 1 بایت

PostgreSQL

INT

SMALLINT: 2 بایت

INTEGER: 4 بایت

BIGINT: 8 بایت

SERIAL

SMALLSERIAL: 2 بایت

SERIAL: 4 بایت

BIGSERIAL: 8 بایت

NUMERIC و DECIMAL

متغیر، بستگی به دقت دارد

FLOAT و DOUBLE

REAL (FLOAT4): 4 بایت

DOUBLE PRECISION (FLOAT8): 8 بایت

CHAR و VARCHAR

CHAR(n): n بایت

VARCHAR(n): متغیر، بسته به طول رشته

TEXT

متغیر، بسته به طول رشته

BYTEA

متغیر، بسته به طول داده

DATE و TIME

DATE: 4 بایت

TIMESTAMP: 8 بایت

بایت 8: TIME

بایت 16: INTERVAL

و reference type کدامیک از نوع tuple و struct و record و enumeration تایپ های هستند stack یا heap و کدامشون value type کدام یک از نوع در زبان های مختلف تفاوت دارند

C#

Enumeration (enum)

Value Type: نوع

Stack: ذخیره سازی

Struct

Value Type: نوع

قرار Heap مگر اینکه بخشی از یک نوع مرجع باشد، در این صورت در Stack: ذخیره سازی (می گیرد)

Record

وجود دارد Record و بالاتر دو نوع C# 9.0 در

Heap ، ذخیره سازی در (Reference Type) نوع مرجع: record

Stack ، ذخیره سازی در (Value Type) نوع مقداری: record struct

Tuple

(به بعد، به عنوان نوع مقداری پیاده سازی شده است C# 7.0 از) Value Type: نوع

Stack: ذخیره سازی

Python

Enumeration (enum)

Reference Type: نوع

Heap: ذخیره سازی

Struct

وجود ندارد. می توان از کلاس ها و دیتاکلاس ها استفاده کرد C# در struct در پایتون ساختاری معادل ذخیره می شوند Heap هستند و در Reference Type که

Record

ندارد، اما می توان از دیتاکلاس ها یا نام گذاری های مختلف استفاده Record پایتون ساختار رسمی برای ذخیره می شوند Heap هستند و در Reference Type کرد که

Tuple

Reference Type: نوع

Heap: ذخیره سازی

heap با stack تفاوت

انواع حافظه در رم Stack و Heap و...

حافظه stack

در بخش space-user 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 memory static 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 Out First In Last(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. حافظه heap

حافظه Heap در قسمت 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 heap متغیر 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 اصطلاح (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.

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