چگونه سایز یک Object را به دست بیاوریم؟

در زبان جاوا بر خلاف زبان های c/c++ متد sizeof() برای بدست آوردن اندازه یک آبجکت (بر اساس بایت)، وجود ندارد. در این مطلب قصد دارم نشون بدم چجوری میشه اندازه آبجکت مدنظرمون رو بدست بیاریم.

مصرف حافظه در جاوا

همونطور که می دونیم در جاوا تمام انواع داده اولیه (primitive data types) اندازه و سایز های استانداردی دارند، و خبری از byte alignment و pad نیست. با این وجود قضیه اونقدر ها هم ساده نیست.

با وجود اینکه انواع داده های اولیه باید طوری رفتار کنند که انگار اندازه های رسمی خودشون رو دارند، یک JVM می تونه دیتا رو هر جوری که می خواد به صورت داخلی (internally) ذخیره کنه، با هر مقدار padding و سر ریز. می تونه انتخاب کنه یک آرایه از Boolean (Boolean[]) رو بصورت 64-bit long ذخیره کنه، یا اینکه یک آبجکت موقت رو روی پشته ذخیره کنه و یا فراخوانی متد ها یا متغیر های رو با متغیرهای ثابت جایگزین کنه و ... . اما برای ما تا زمانی که برنامه نتیجه یکسانی بده اهمیتی نداره.

همینطور در نظر بگیریم که حافظه cache سخت افزار و سیستم عامل هم موثره پس در واقع ما تنها می تونیم یک پیش بینی تقریبی از مصرف رم رو داشته باشیم.

اشیا، ارجاعات و کلاس های در برگیرنده(Wrapper)

آبجکت ها در JDK مدرن 64 بیتی حداقل اندازه ی 16 بایت دارند. از اونجایی که هدر آبجکت 12 بایت دارد، به 8 بایت چندتایی پد میشه. در JDK 32 بیتی، سر ریز 8 بایت است که به 4 بایت چندتایی پد می شود.

ارجاعات در پلتفرم 32 و 64 بیتی 4 بایت حجم دارند، با محدوده حافظه heap کمتر از 32 گیگابایت (-Xmx32G) و 8 بایت اگر این محدوده بیشتر از 32گیگابایت باشه.

این یعنی یک JVM 64بیتی 30-50% فضای هیپ بیشتری نیاز داره.

نکته ای که اینجا وجود داره این هست که انواع box، آرایه ها، رشته ها و سایر ظرف ها مثل آرایه های چند بعدی برای حافظه هزینه بر هستند (memory-costly) چون سرریز مشخصی دارند. برای مثال، وقتی داده اولیه int (که 4 بایت فضا می گیره) رو با با نوع Integer (که 16 بایت فضا لازم داره) مقایسه کنیم، متوجه می شیم که 300% سرریز حافظه داریم.

تخمین اندازه یک آبجکت با استفاده از Instrumentation

یک روش اندازه گیری سایز یک آبجکت استفاده از متد getObjectSize(object) از واسط Instrumentation که از جاوا 5 وجود داره هست.

همونطور که در جاوا داک اومده، این متد "تخمینی بر اساس پیاده سازی" برای سایز آبجکت مورد نظر فراهم می کنه. توجه کنید که نتیجه گیری در اندازه سر ریز و مقادیر در فراخوانی یک JVM سینگل می تونه متفاوت باشه.

این روش تنها تخمینی از سایز آبجکت در نظر گرفته شده به تنهایی رو نشون میده نه اشیایی که آبجکت به اون ها ارجاع می ده. برای تخمین اندازه کلی آبجکت نیاز به کدی داریم که سایز ارجاعات دیگر رو اندازه گیری کنه.

ساخت Instrumentation Agent

public class InstrumentationAgent {

    private static volatile Instrumentation globalInstrumentation;

    public static void premain(final String agentArgs, final Instrumentation inst) {

        globalInstrumentation = inst;

    }

    public static long getObjectSize(final Object object) {

        if (globalInstrumentation == null) {

            throw new IllegalStateException("Agent not initialized.");

        }

        return globalInstrumentation.getObjectSize(object);

    }

}

کلاس نمونه

public class InstrumentationExample {

    public static void printObjectSize(Object object) {

        System.out.println("Object type: " + object.getClass() +

          ", size: " + InstrumentationAgent.getObjectSize(object) + " bytes");

    }

    public static void main(String[] arguments) {

        String emptyString = "";

        String string = "Estimating Object Size Using Instrumentation";

        String[] stringArray = { emptyString, string, "com.baeldung" };

        String[] anotherStringArray = new String[100];

        List<String> stringList = new ArrayList<>();

        StringBuilder stringBuilder = new StringBuilder(100);

        int maxIntPrimitive = Integer.MAX\_VALUE;

        int minIntPrimitive = Integer.MIN\_VALUE;

        Integer maxInteger = Integer.MAX\_VALUE;

        Integer minInteger = Integer.MIN\_VALUE;

        long zeroLong = 0L;

        double zeroDouble = 0.0;

        boolean falseBoolean = false;

        Object object = new Object();

        class EmptyClass {

        }

        EmptyClass emptyClass = new EmptyClass();

        class StringClass {

            public String s;

        }

        StringClass stringClass = new StringClass();

        printObjectSize(emptyString);

        printObjectSize(string);

        printObjectSize(stringArray);

        printObjectSize(anotherStringArray);

        printObjectSize(stringList);

        printObjectSize(stringBuilder);

        printObjectSize(maxIntPrimitive);

        printObjectSize(minIntPrimitive);

        printObjectSize(maxInteger);

        printObjectSize(minInteger);

        printObjectSize(zeroLong);

        printObjectSize(zeroDouble);

        printObjectSize(falseBoolean);

        printObjectSize(Day.TUESDAY);

        printObjectSize(object);

        printObjectSize(emptyClass);

        printObjectSize(stringClass);

    }

    public enum Day {

        MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY

    }

}

منبع: https://www.baeldung.com/java-size-of-object