

موضوع انتخاب شده: Automotive

روند کلی کار:

در ابتدا ما پوشه Automotive را از سایت mibench دریافت کردیم و تمام فایل‌های داخل پوشه‌های آن را کامپایل نمودیم. ابتدا کد اسمبلی تمامی این فایل‌ها را به دست آوردیم که طریقه استخراج تعداد دستورات موجود در آن در ادامه آمده، سپس زمان اجرای هر برنامه را (توابع داخل آن) را برای پیدا کردن گلوگاه برنامه به دست آوردیم و بعد از اینکه تابع‌های گلوگاه مشخص شد داخل کد C یک متغیر برای محاسبه مدت زمان اجرای کل آن تابع و یک متغیر برای به دست آوردن مدت زمان اجرای دستورات داخل تابع قرار دادیم و این دو را با هم مقایسه کردیم. هر بار که گلوگاه درون تابع پیدا می‌شد ما تعداد دستورات کمتری را بررسی می‌کردیم (یعنی فاصله بین شروع و پایان کلاک را کوچک‌تر می‌کردیم) تا بتوانیم به صورت دقیق یک یا دو دستوری که باعث زیاد شدن زمان اجرای برنامه می‌شوند را پیدا کنیم که در اکثر موارد به تقسیم، ضرب و توان می‌رسیدیم که البته تقسیم در برنامه‌های انتخابی توسط ما زمان بیشتری را به خود اختصاص می‌داد.

بررسی برنامه‌ها به صورت تکی:

: Susan

ورودی این برنامه بسیار کم حجم بود و برای اینکه بتوانیم مدت زمان اجرای توابع را به صورت دقیق‌تر بررسی کنیم یک ورودی با حجم بالاتر به برنامه دادیم و خروجی `susan.txt` را گرفتیم. سپس پس از مشخص شدن تابع گلوگاه داخل این برنامه کلاک‌های کل و کلاک عملیاتی که به گلوگاه بودن آن مشکوک شده بودیم بررسی کردیم و حدسمان درست بود و اکثر زمان را به خود اختصاص می‌داد (اسکرین شات داخل پوشه). در این برنامه عملیات تقسیم به تنهایی زمان بسیار زیادی به خود اختصاص می‌داد به همین دلیل این عمل به عنوان گلوگاه برنامه ما شناخته شد.

: Cubic

در این برنامه هم مانند Susan عمل کردیم و یک تابع خاص زمان بسیار زیادی را به خود اختصاص می داد و داخل یک حلقه اجرا می شد. به داخل آن تابع که در یک فایل دیگر بود رفتیم و با بررسی کلاک ها متوجه شدیم یک **for** زمان زیادی به خود اختصاص می دهد که شامل عملیات تقسیم، توان و ضرب است که ما تقسیم را انتخاب کردیم.

در این برنامه **for** ای که ما انتخاب کردیم نصف مدت زمان اجرای تابعی را می گرفت که همان تابع 66 درصد مدت زمان اجرای کل برنامه را شامل می شد.

: Bin-to-dec

این برنامه را از گیت هاب پیدا و **clone** کردیم که شامل یک فایل **C** می شد .

داخل این برنامه یک حلقه که داخل آن عمل % و تقسیم انجام می شد بیشترین زمان برنامه را می گرفت که ما عمل تقسیم را انتخاب کردیم.

توضیحات فایل برنامه:

- 1) فایل های **C**. سورس کد برنامه می باشند که آن ها را کامپایل می کنیم.
- 2) فایل با فرمت **S**. شامل فایل اسمبلی خام برنامه می باشد.
- 3) **Screenshot** نشان دهنده بررسی دقیق تر مدت زمان اجرای دستورات در برنامه می باشد که با اندازه گیری تعداد کلاک ها در شروع و پایان انجام می شود.
- 4) یک فایل **txt**. برای مشخص کردن مدت زمان اجرای توابع و مقایسه آنها در برنامه وجود دارد.

توضیحات تحلیل دستورات در کد اسمبلی :

فایل های که از **disassemble** به دست می آیند با پسوند **S** هستند. پسوند آنها را به **txt** تغییر داده و سپس متن را به نرم افزار ورد منتقل کرده و به صورت خودکار داری نظمی مشخص خواهند شد.

برنامه ای در جاوا نوشته و طبق نظم موجود دستورات را از یکدیگر جدا کرده و تعداد استفاده آنها محاسبه می شود. وظیفه دیگر این برنامه مرتب کردن داده ها و دستورات برای رسم نمودار فراوانی است. در پایان دستورات به همراه تعداد کاربرد آنها به صورت نزولی چاپ می شوند.

در نهایت ما عمل تقسیم را برای پیاده سازی سخت افزاری در فازهای بعد انتخاب کردیم.