

MohammadAmin Shafiee  
(9727713)

## Classes:

ir.ac.kntu.os.Tree.java:

پیاده سازی یک درخت است برای پیدا کردن یک بلاک مناسب. برای تخصیص به یک فرایند.

ir.ac.kntu.os.Block.java:

یک بلاک برای که دارای سایز است. در هر بلاک الگوریتم اولین برازش برای پیدا کردن ادرس مورد نظر استفاده می شود.

ir.ac.kntu.os.OsMemoryManagement.java:

سیستم بادی در این کلاس پیاده سازی شده است همراه با مرج کردن بلاک های آزاد.

ir.ac.kntu.os.MemoryReporter.java:( singleton design pattern)

یک ریسمان است. که هر ۵ ثانیه یک بار تمام اطلاعات ریسمان های دیگر را و همچنین اطلاعات مموری را در قالب  
Json  
در ترمینال نشان میدهد هم در یک فایل با توجه به زمان در فولدر  
Log  
دخیره می کند.

Ir.ac.kntu.os.Locker.java(singleton design pattern)

درای تمام قفل های مشترک بین فرایندها است. دارای ۳ تا قفل برای درخت، لیستی که دارای بلاک های استفاده شده است و همچنین قفلی برای متغیری که نشان می دهد همه ی فرایندها تمام شده اند یا نه.

Ir.ac.kntu.process.Process.java:

همان ریسمان هایی هستند که درخواست گرفتن و آزاد کردن بلاک را می دهند.

Ir.ac.kntu.process.ProcessRandomGenerator.java:

برای زمان خواب به صورت رندم برای هر فرایند و درخواست سایز بلاک به صورت رندم و همچنین درخواست رندم استفاده می شود. درخواست رندم به صورت احتمال وزن دار است برای زیر ۳۲ کیلو بایت میزان وزن بیشتر است.

ir.ac.kntu.UI:

این پکیج دارای ریسمان ها و قالب اصلی نمایش است. برای کشیدن چارت و درخت.

موارد امتیازی:

1. UI

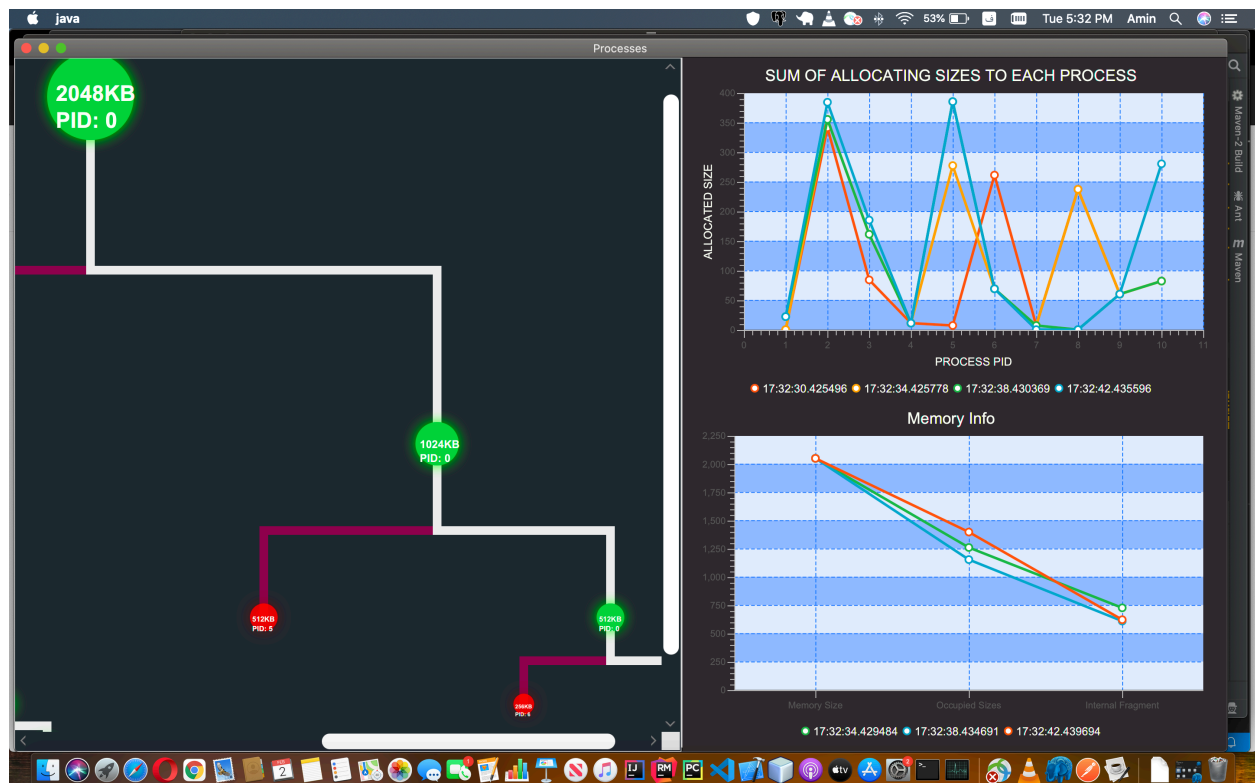
2. Merging Freed Blocks:

به هم وصل کردن بلاک های آزاد

3.first\_fit Algorithm

در هر بلاک first\_fit پیاده سازی الگوریتم

Outputs:



فایل json که اطلاعات مموری را نگه می دارد در همین زمان:

2021-02-02T17:32:42.265728.json

```
{
  "Internal Fragment": "620KB",
  "total space used": "1396KB",
  "total free spaces": "652KB",
  "Processes": [
    {
      "Addresses": [
        128
      ],
      "Occupied Spaces": 22,
      "Is Process Finished": false,
      "Process Id": 1,
      "Start Time": "2021-02-02T17:32:07.124783",
      "Execution Time": "00:00:35.000000140"
    },
    {
      "Addresses": [
        384,
        512,
        668,
        736,
        96
      ],
      "Occupied Spaces": 384,
      "Is Process Finished": false,
      "Process Id": 2,
      "Start Time": "2021-02-02T17:32:07.125004",
      "Execution Time": "00:00:35.000000140"
    },
    {
      "Addresses": [
        256,
        340,
        192,
        0
      ],
      "Occupied Spaces": 185,
      "Is Process Finished": false,
      "Process Id": 3,
      "Start Time": "2021-02-02T17:32:07.125025",
      "Execution Time": "00:00:35.000000140"
    }
  ],
}
```

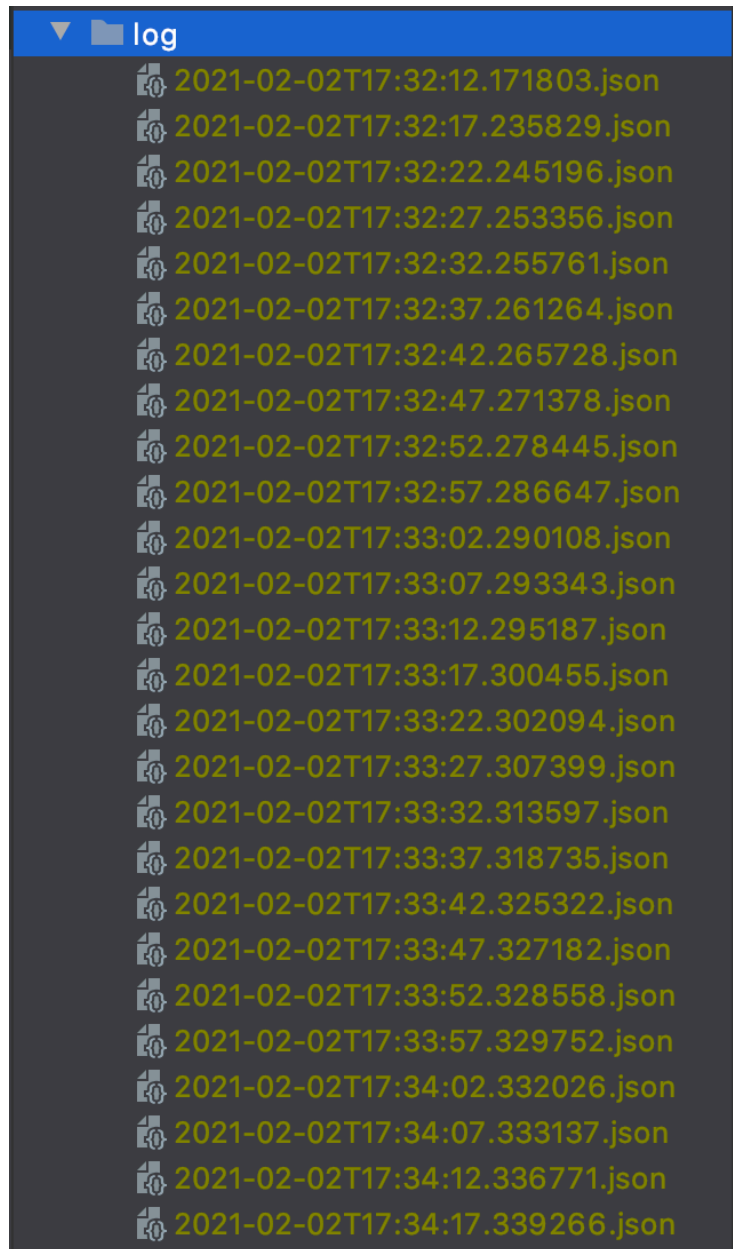
```
{
  "Addresses": [
    80
  ],
  "Occupied Spaces": 11,
  "Is Process Finished": false,
  "Process Id": 4,
  "Start Time": "2021-02-02T17:32:07.125041",
  "Execution Time": "00:00:35.000000140"
},
{
  "Addresses": [
    160,
    1024,
    1294
  ],
  "Occupied Spaces": 385,
  "Is Process Finished": false,
  "Process Id": 5,
  "Start Time": "2021-02-02T17:32:07.125055",
  "Execution Time": "00:00:35.000000140"
},
{
  "Addresses": [
    704,
    320
  ],
  "Occupied Spaces": 69,
  "Is Process Finished": false,
  "Process Id": 6,
  "Start Time": "2021-02-02T17:32:07.125096",
  "Execution Time": "00:00:35.000000140"
},
{
  "Addresses": [],
  "Occupied Spaces": 0,
  "Is Process Finished": false,
  "Process Id": 7,
  "Start Time": "2021-02-02T17:32:07.125131",
  "Execution Time": "00:00:35.000000140"
},
}
```

```

    {
      "Addresses": [],
      "Occupied Spaces": 0,
      "Is Process Finished": false,
      "Process Id": 8,
      "Start Time": "2021-02-02T17:32:07.125155",
      "Execution Time": "00:00:35.000000140"
    },
    {
      "Addresses": [
        128
      ],
      "Occupied Spaces": 60,
      "Is Process Finished": false,
      "Process Id": 9,
      "Start Time": "2021-02-02T17:32:07.125176",
      "Execution Time": "00:00:35.000000140"
    },
    {
      "Addresses": [
        104,
        256,
        768
      ],
      "Occupied Spaces": 280,
      "Is Process Finished": false,
      "Process Id": 10,
      "Start Time": "2021-02-02T17:32:07.125197",
      "Execution Time": "00:00:35.000000140"
    }
  ]
}

```

فولدر log بعد از تمام شدن برنامه:



اجرا:

```
mvn clean javafx:run
```

ضمیمه:

در پوشه ScreenShotsAndVideo چندین عکس و video برای نشان دادن عملکرد آمده است.

در آخر من log اجرای قبلی را در فولدر log می گذارم شما می توانید آن را به عنوان نمونه مشاهده کنید.