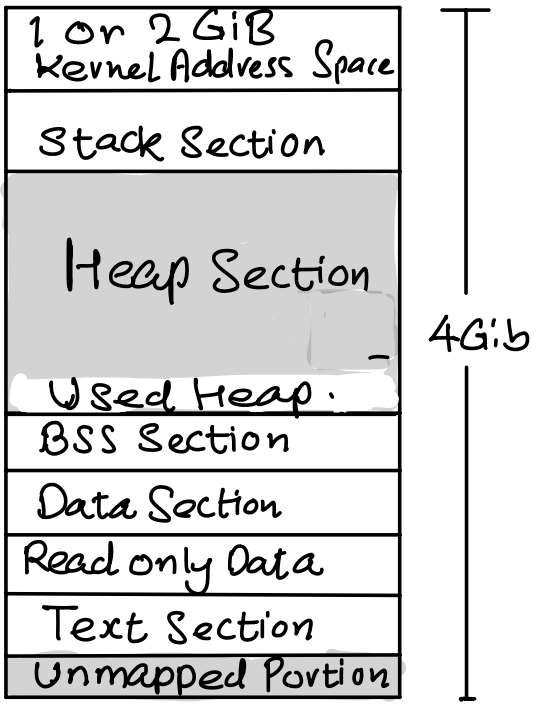
Memory Model Provided by 32-bit O.S. to a process.

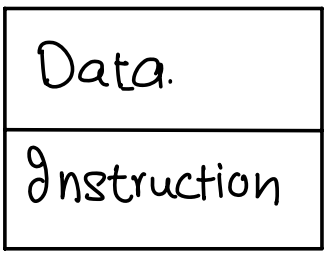
a.k.a. Virtual Address Space Layout of a process.



total V.A. = 4Gib(32bit) − 1,2 Gib (Kernel Address Space) = 256 Tib(64bit) − 1,2Gib

User Address Space.

O.S. ⟶ app. User Address Space.

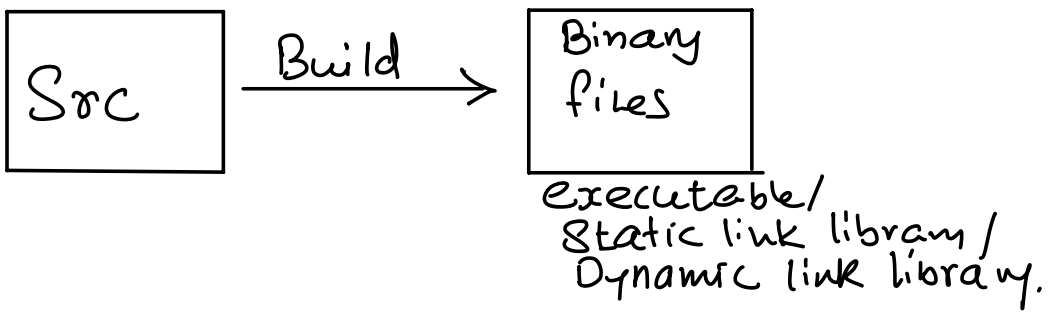


- User Address Space : text : Instructions stored in Machine's Language.
- Data:
  1) Read only Data Section.
  2) Data Section.
  3) BSS section
  4) Stack section.
  5) Heap Section.
- Data Allocation Methods.
  1) Data Definition Statement.
  2) Call Memory Allocation Functions.
- Data Definition Statement
  - Global Scope
    - Initialized
    - Uninitialized.
  - Local Scope.

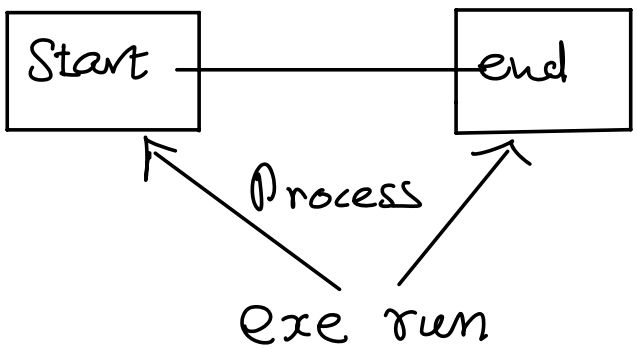
- 2 kind of times.

1) Static time:

Time elapsed while building application.



2) Dynamic time / Run time.



---

1) Data manipulation statements of all functions in source code ⟶ Text Section.
2) Global Initialized D.D.S. ⟶ Data Section.
3) Global Uninitialized D.D.S ⟶ BSS section.
4) Local Data Definition Statements ⟶ Stack.
5) Memory Allocation function ⟶ Heap.
6) Global D.D.S. Qualified by const:
   Local D.D.S. Qualified by Static + const:
   Strings. = Read Only Data Section.

Src ——> exe.

text / rodata / Data / BSS. ——> exe.

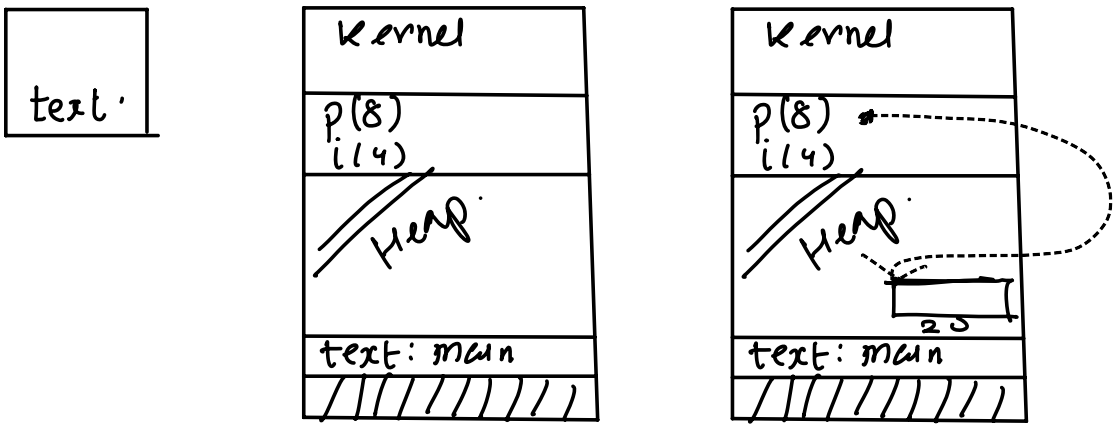
Static Sections.

```
int main(void){
    int* p = NULL;
    int i;

    p = (int*)malloc(5 * sizeof(int));
    assert(p != NULL);
    for(i = 0; i < 5; ++i)
        *(p+i) = (i+1) * 10;

    free(p);
    p = NULL;
    exit(0);
}
```

```
malloc_demo.c    -> # gcc -o app malloc_demo.c
                    # cl /Fe:app.exe malloc_demo.c
```



1) Advanced Programming Under Unix Environment (3rd edition) -> Stevens, Rago
2) Advanced C/C++ Compiling -> Apress -> compiler, assembler, linkers
3) Advanced 80386 Programming -> James Turley (first 4 chapters)
4) Understanding the Linux Kernel (3rd Edition)-> Chapter 2 -> Memory addressing
   Chapter 8, 9 -> Memory Management Chapters
   (Cessati, Bovet)