

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY, VNU HCM  
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



# OPERATING SYSTEM

---

## Lab 4: Process (Part 2/2)

---

Pham Minh Tuan - MSSV: 1752595



## 1 Problem 2:

Write a short essay which summarizing the knowledge of process's data segment to answer for these questions:

- In which cases we should use `aligned_malloc()` instead of standard `malloc` ?
- How can we increase the size of heap in a running process ?

We use `aligned_malloc()` to allocate memory to aligned directory that is the certain bit boundary. Since the standard `malloc` will return the suitable alignment for any standard types depending on the OS architecture, it sometimes not useful whenever we allocate too much memory space or too low one. So the `aligned_malloc()` will solve that problem. If we know the size and the aligned bits boundary, the program can run faster and efficiency. For UNIX system, the `malloc` is implemented by **glibc**, the reason we need to use `aligned_malloc()` will be more clear:

*The address of a block returned by `malloc` or `realloc` in GNU systems is always a multiple of eight (or sixteen on 64-bit systems). If you need a block whose address is a multiple of a higher power of two than that, use `aligned_malloc()` or `posix_memalign`*

We can increase the size of heap whenever we call `malloc()`. For UNIX system, with the `brk()` and `sbrk()`, we can adjust the program break as long as the virtual memory allocated enough for it. We typically don't need to call `brk()` everytime if `malloc` provide enough memory otherwise we can call `setbrk()` system call. The limit of the size of virtual memory of the system