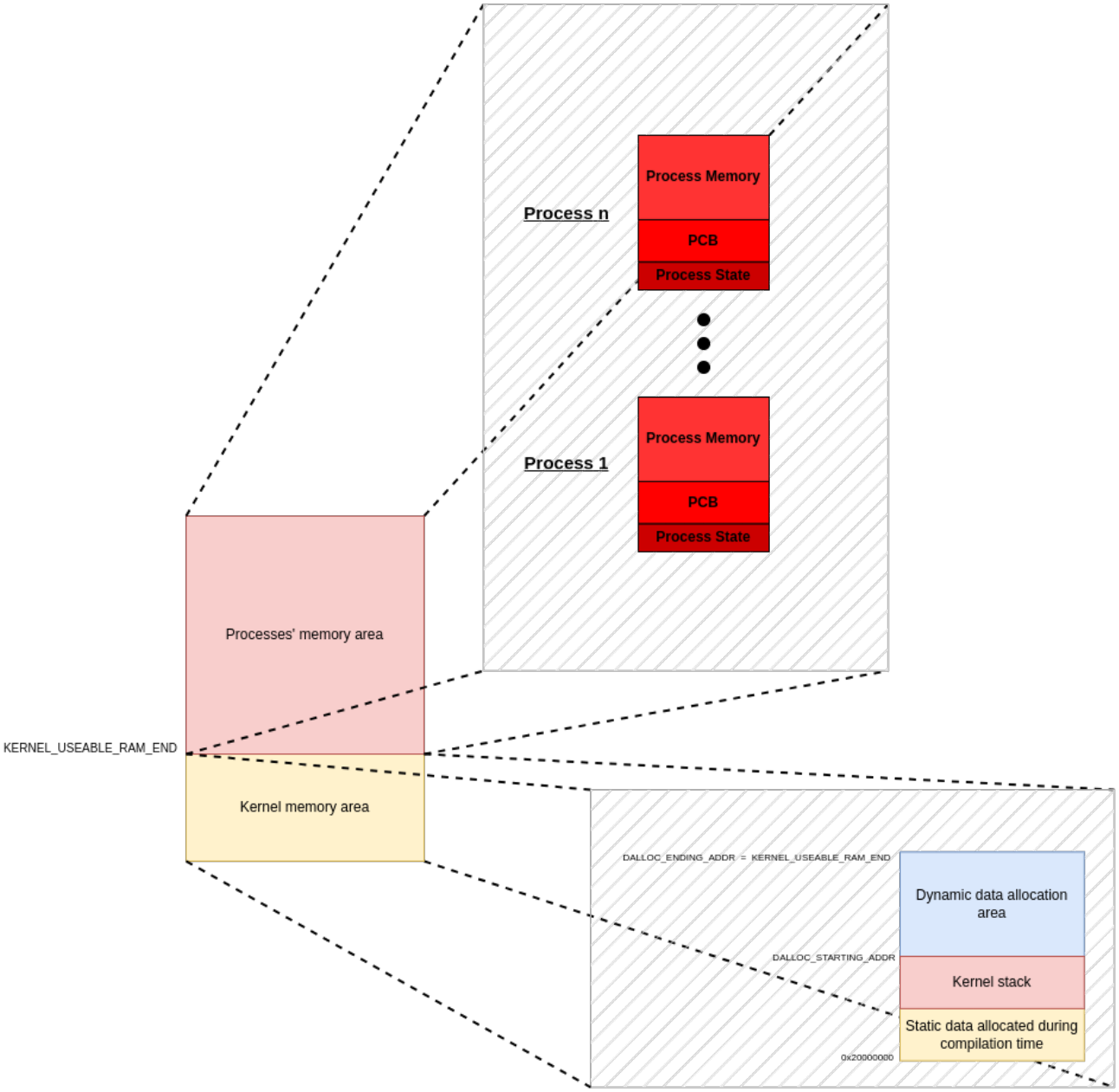
Chapter:
Memory organization by the OS

In this chapter we are going to analyze the way the memory is manipulated by the Operating System not only from the perspective of the kernel, but also from the perspective of the process.

General memory organization picture

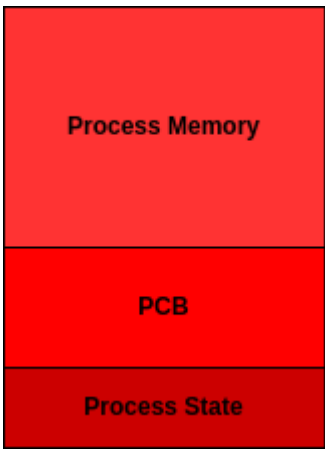
The following picture shows how the memory is treated by the Operating System



Those structures depicted in the figure are going to be clarified soon one-by-one.

How process-pages work

The total memory has a whole area dedicated for posible process that might get attached to the scheduler. This area is divided into fixed-size pages, called process-pages. Each process page has the following format:

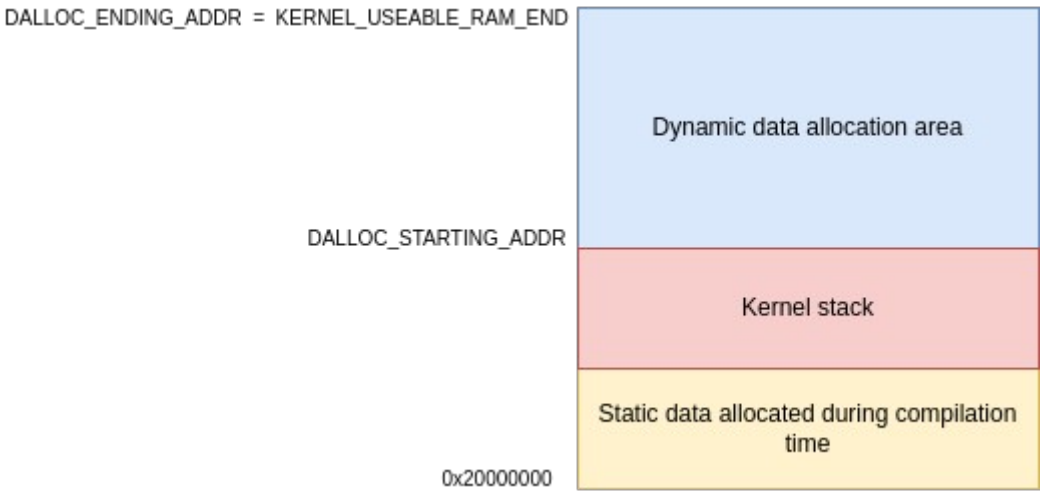


where

- **Process State**: Indicates whether this process-page is in use or not
- **PCB**: holds useful data like register values; needed for the context-switching
- **Process Memory**: the memory area handled by the process

How kernel-specific memory is organized

The term “kernel-specific memory” refers to the memory area that the kernel uses to implement its basic tasks e.g. hold a queue for the running processes or hold current running process usesful data. This area is depicted below, with the labels to the left having the same names as the macros used in source code for this purpose



How dynamic allocation works

In order to have the ability of dynamically allocating data and to avoid problems like fragmentation, we accompany the data to be dynamically allocated with meta-data, which store the size of the data-area to be allocated. In fact, the data that we store has the following picture:

In order for dynamic allocation to work, the dynamic data allocation area must be initialized to zeroes everywhere. Whenever malloc() (which is implemented from scratch) is called, it tries to find an empty slot where it can store the above frame. It is obvious that, when data is to be allocated, the first two bits are going to be non-zero because the size of the data can never be zero. That's why we are sure that in a memory area with only zeroes is likely to host the data. A final check must be made to assure that the zeroes are sufficient and the data fit-in there.