

Process concept

Concept of Multiprogramming

- When there is a single program running in the CPU, it leads to the degradation of the CPU utilization.
- Example: When a running program initiates an I/O operation, the CPU remain idle until the I/O operation is completed.

Solution to this problem is provided by
Multiprogramming.

Multiprogramming Continued..

Definition:

A mode of operation that provides for the interleaved execution of two or more programs by a single processor.

Multiprogramming Continued..

Improving CPU utilization

By allowing several programs to reside in main memory at the "**same time**" the CPU might be shared,

such that when one program initiates an I/O operation,

another program can be assigned to the CPU, thus the improvement of the CPU utilization.

Multiprogramming Continued..

Implementation

The concept of ***process*** need to be introduced in order to understand multiprogramming and the concurrent execution of sequential processes in a computer system.

What is a process?

Process

■ Definition:

- A program in execution
- An asynchronous activity
- The “locus of control” of a procedure in execution
- It is manifested by the existence of a process control block (PCB) in the operating system.

Process States

A state of a process describes the activity that the process is doing at a certain moment in time.

New : A newly created process, not in the ready queue.

Ready : It can use the CPU , if available.

Running : If it is currently in the CPU.

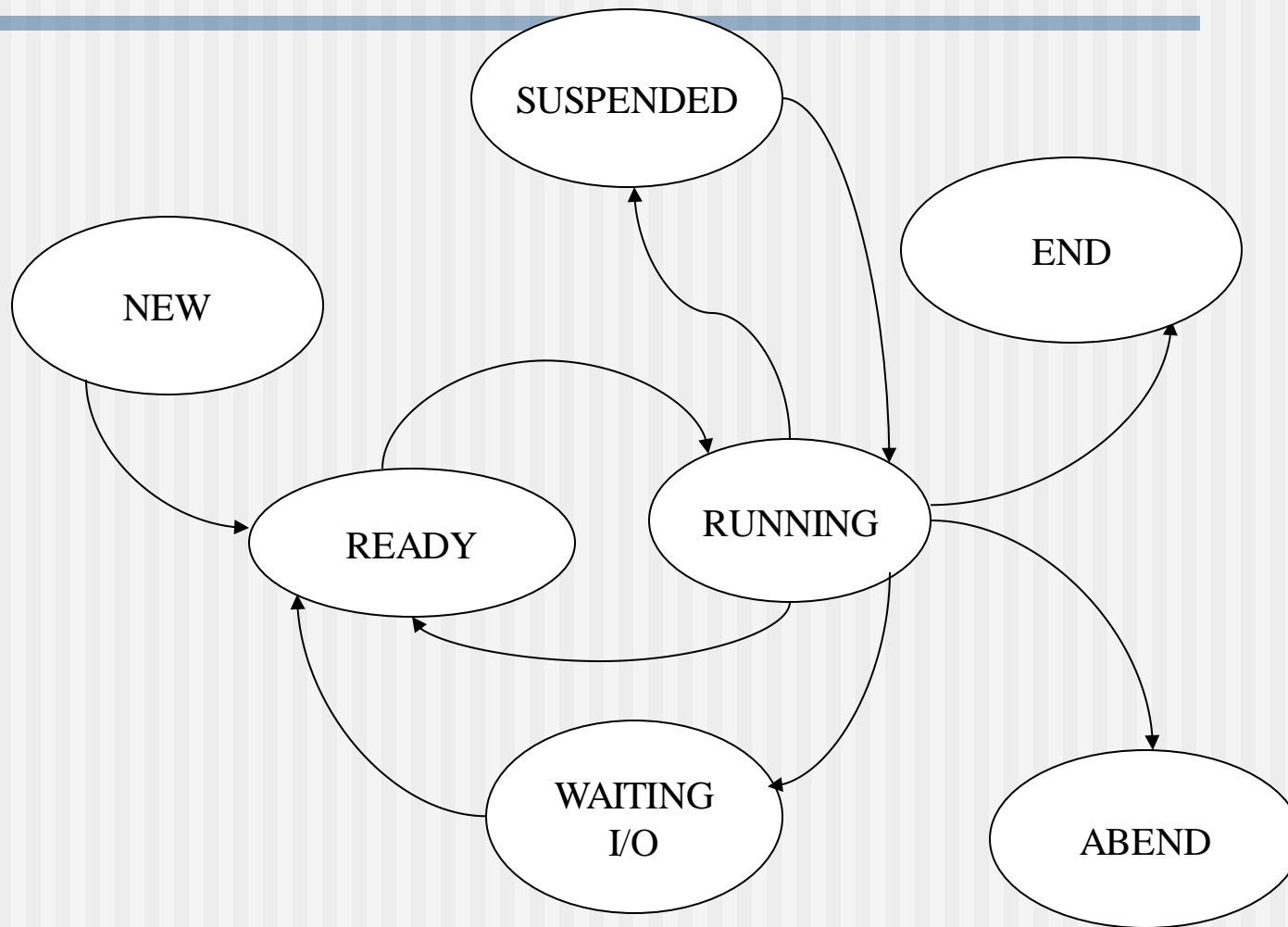
Waiting : Waiting for some event ex: I/O

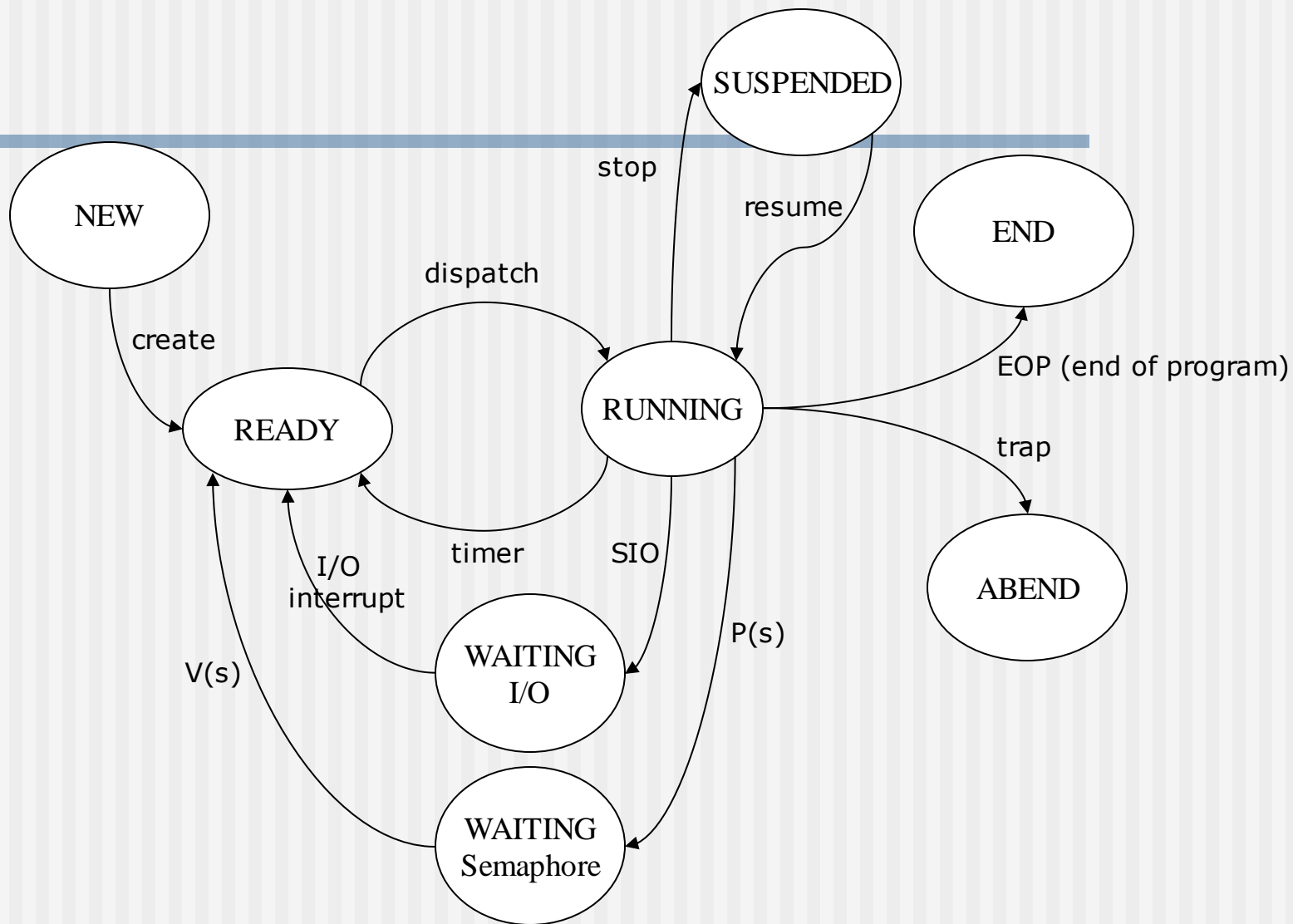
Abend : Stops executing due to an error.

End : Finished executing properly.

Suspended: Stops temporarily while the OS uses the CPU

States of processes



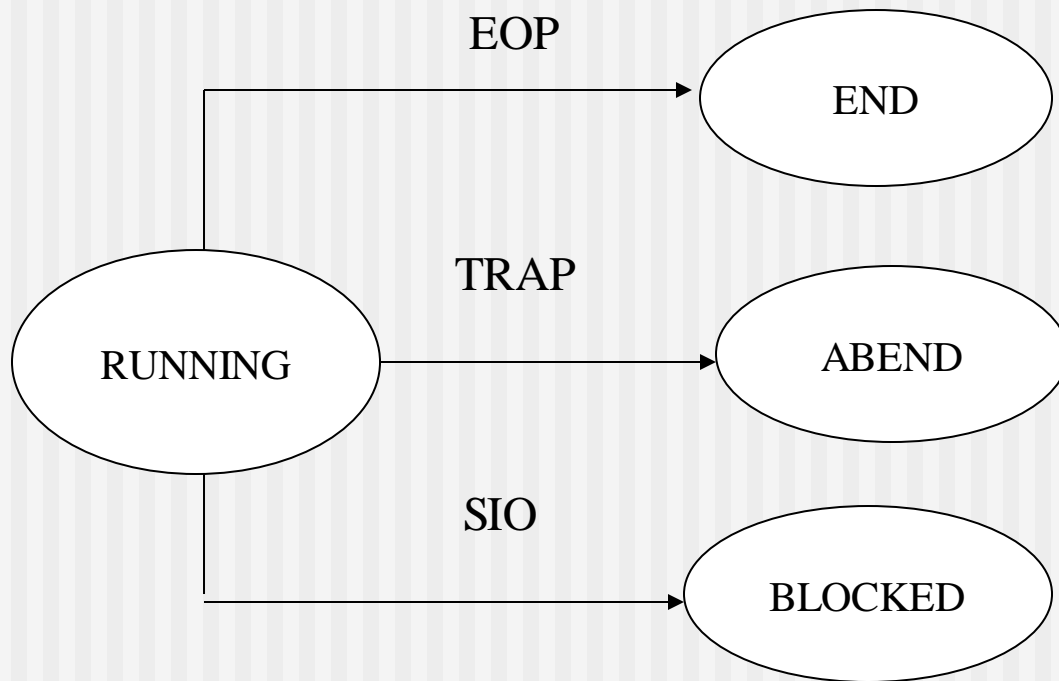


Causes of state change

When a process executes, it changes states and interrupts cause process to change states.

<u>Current State</u>	<u>New state</u>	<u>Interrupt</u>
Running (End of Program)	End	EOP
Running (Abnormal end)	ABEND	Trap
Running (Start I/O)	Blocked for I/O	System Call (SIO)

Depiction of state change



Process Continued...

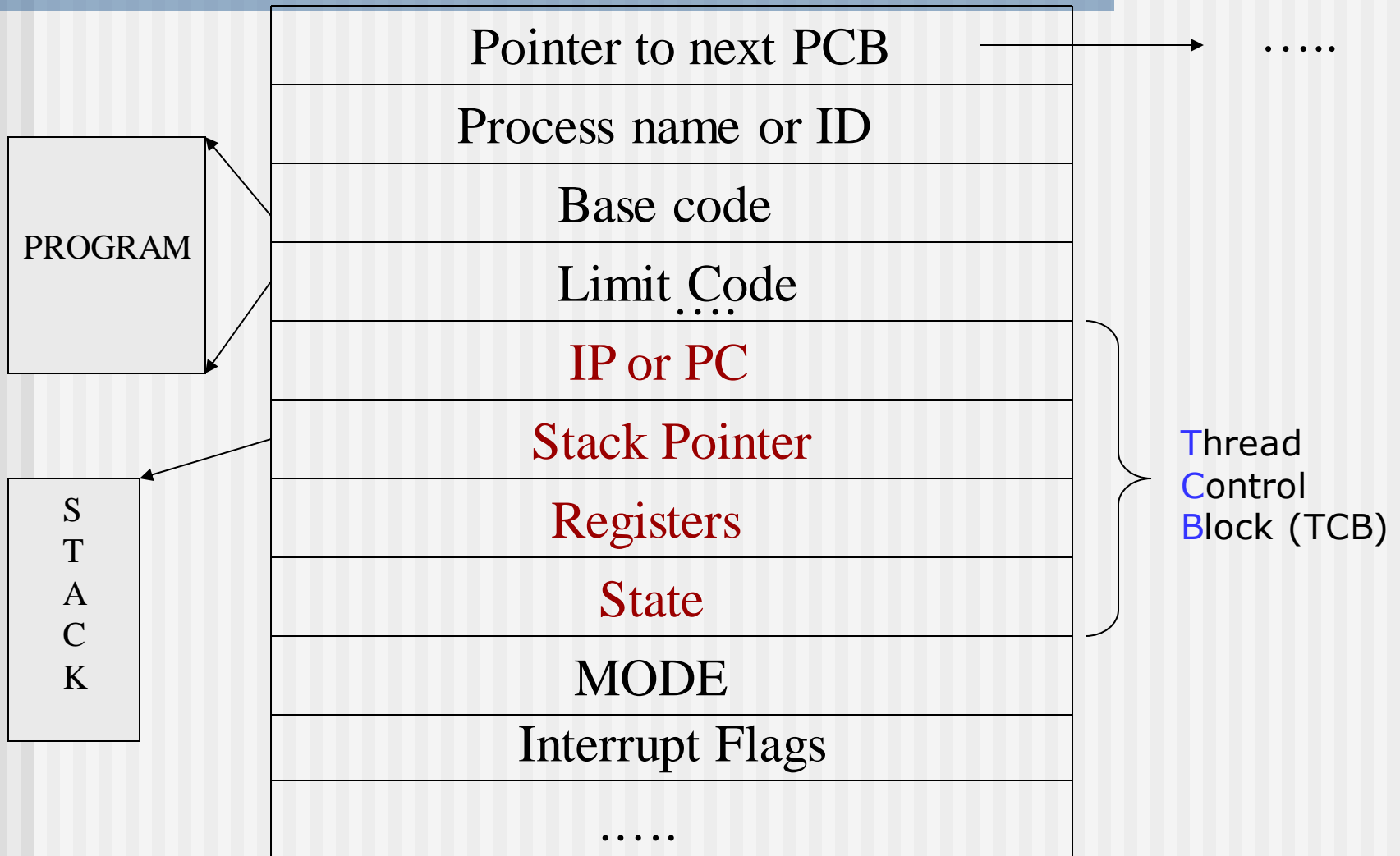
- The activity of a process is controlled by a data structure called **P**rocess **C**ontrol **B**lock(PCB).
- A PCB is created every time a program is loaded to be executed.
- So, a process is defined by PCB-Program couple.

Structure of PCB

PCB contains information about processes, for instance:

- the current state of a process
- Unique identification of process
- Process priority
- Contents of some CPU registers
- Instruction Pointer (IP), also known as PC
- Base and limit registers
- Time limits and I/O status information

Structure of PCB Contd...



Process Continued...

A thread is known as “the unit of execution” of a process and it is represented by the Thread Control Block (TCB).

The TCB consists of four fields: PC, stack pointer, the contents of some registers, and the state.

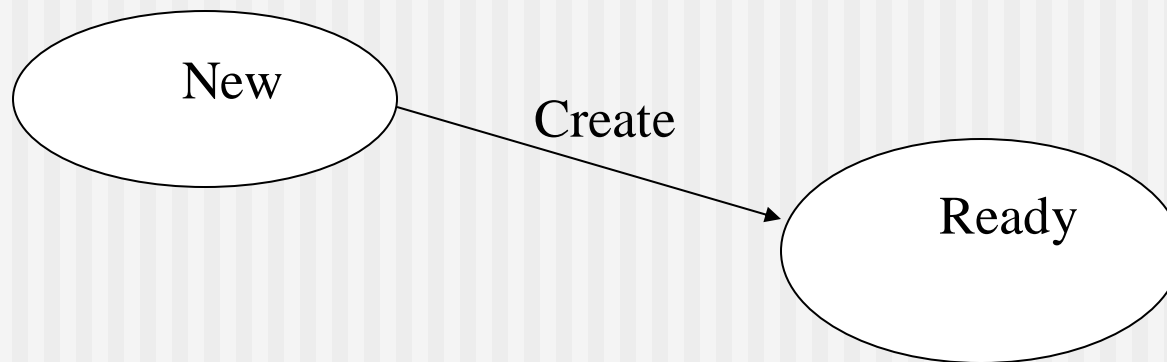
Process Continued...

We can now observe how each stage of a process takes place by the aid of a state diagrams.

Process creation :

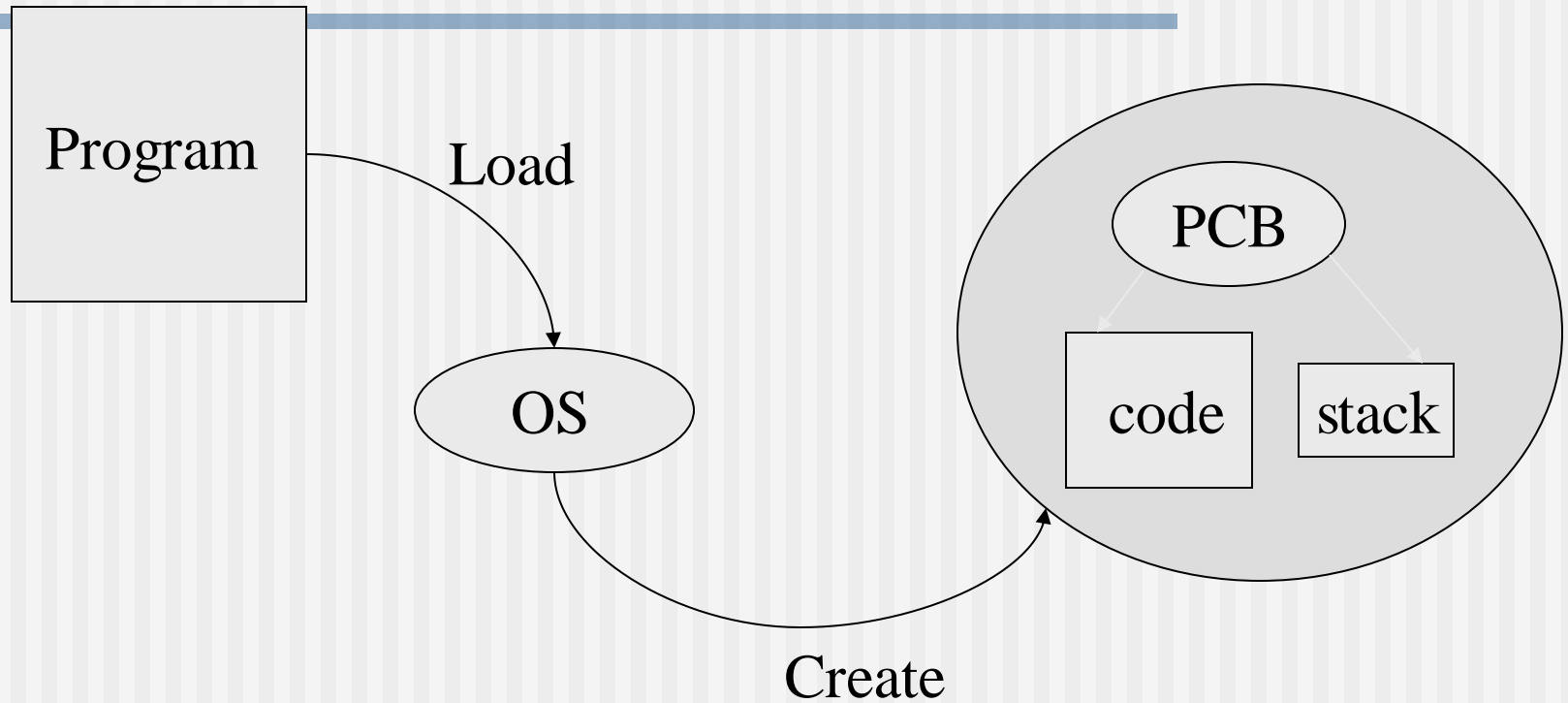
- An OS can create one or more processes, via a create-process system call.
- During the course of execution an user process may create new processes or threads as well. In this case, the creating process (thread) is called the parent and the created (new) process (thread) is named the child.

Process Creation

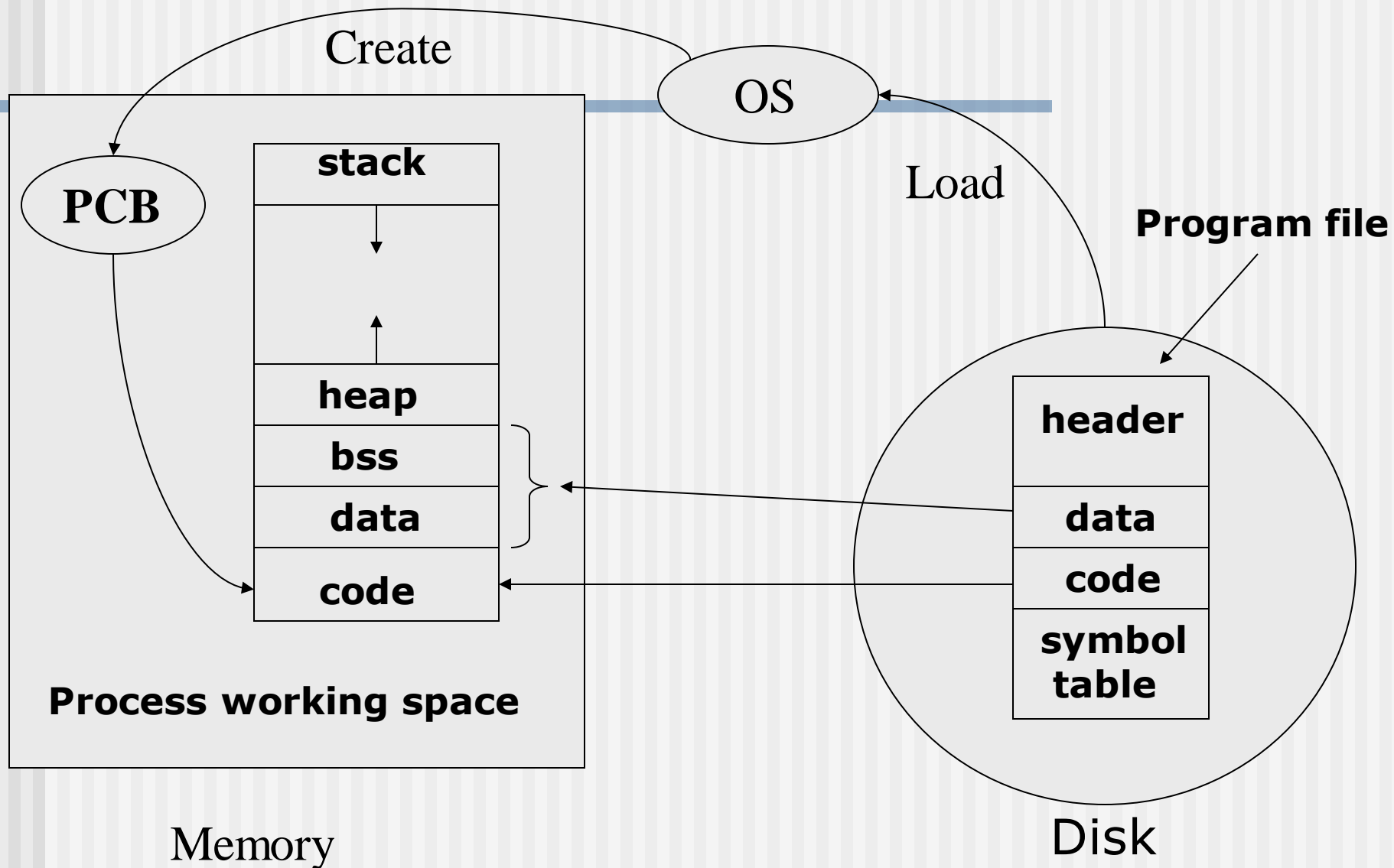


The process is created and then inserted at the back of the ready queue, it moves to the head of the queue according to a scheduling policy.

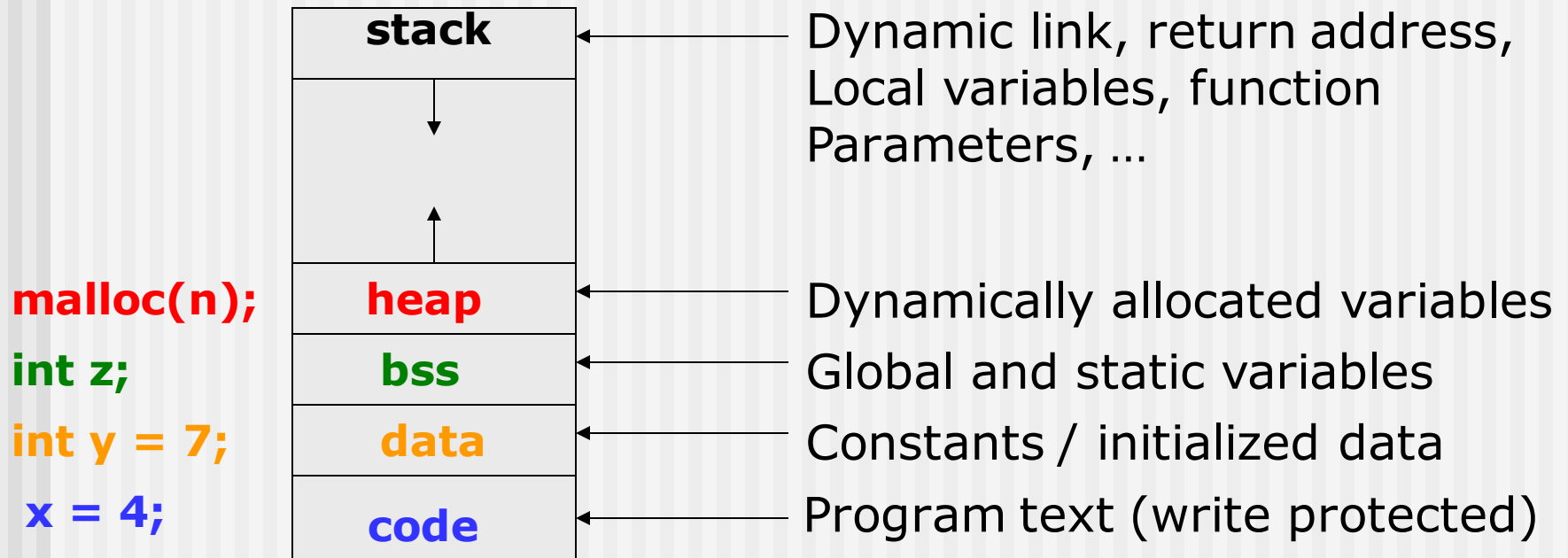
Process Creation Contd...



Process Creation (closer view) Cont...



Process working space (run-time environment)



Process working space

Process working space (run-time environment)

Process working space

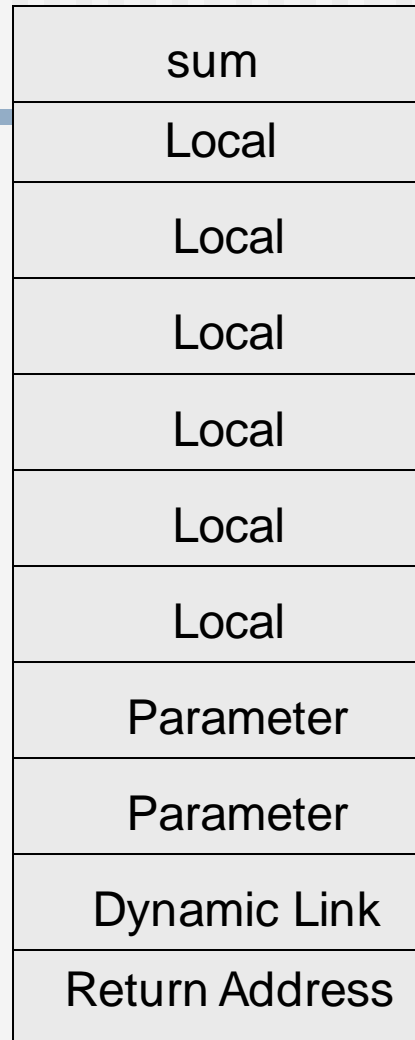
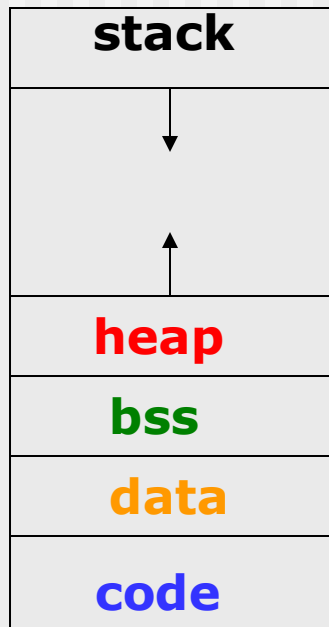
bss: means "block started by symbol" and has that name for historical reasons.

malloc(n);

int z;

int y = 7;

x = 4;

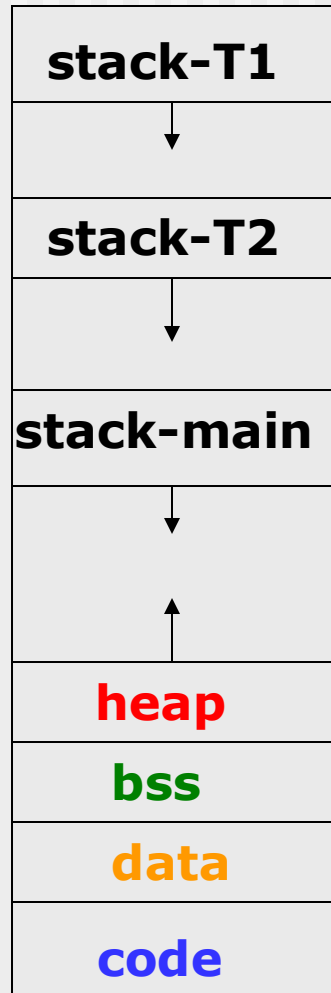


```
void sub(float total, int part ) {  
  int List[5];  
  float sum;  
  ...  
}
```

← **Activation record**

Thread working space (run-time environment)

Multithreading: Each thread is a unit of execution.



Multithreaded processes need a stack per thread.

All threads shared the same address space.

Each thread has its own TCB.

Thread working space (run-time environment)

Multithreading: All threads share the CPU-PC and CPU-SP.
In this picture thread-T1 is using the CPU.

