1. Indicate whether the following statements are true or false. Justify your answers.

a) A ready process waiting to get access to the CPU is in the “waiting” state.

b) A ready queue is a queue of Process Control Blocks (PCBs) of all processes in the “ready” state.

c) The “wait()” system call is generally used by a child process to wait for instructions from a parent process.

d) Message passing based Inter-Process Communication (IPC) consumes less memory than shared memory based IPC.

1a) A ready process waiting to get access to the CPU is in the “waiting” state.

→ False.

Justification: The process is in the “ready” state. A process in the “waiting state” is waiting for an event or the completion of an I/O operation.

1b) A ready queue is a queue of Process Control Blocks (PCBs) of all processes in the “ready” state.

→ True.

Justification: The short-term scheduler uses this queue to schedule processes. Usually, the head of the queue contains the PCB of the process that is currently in the “running” state.

1c) The “wait()” system call is generally used by a child process to wait for instructions from a parent process.

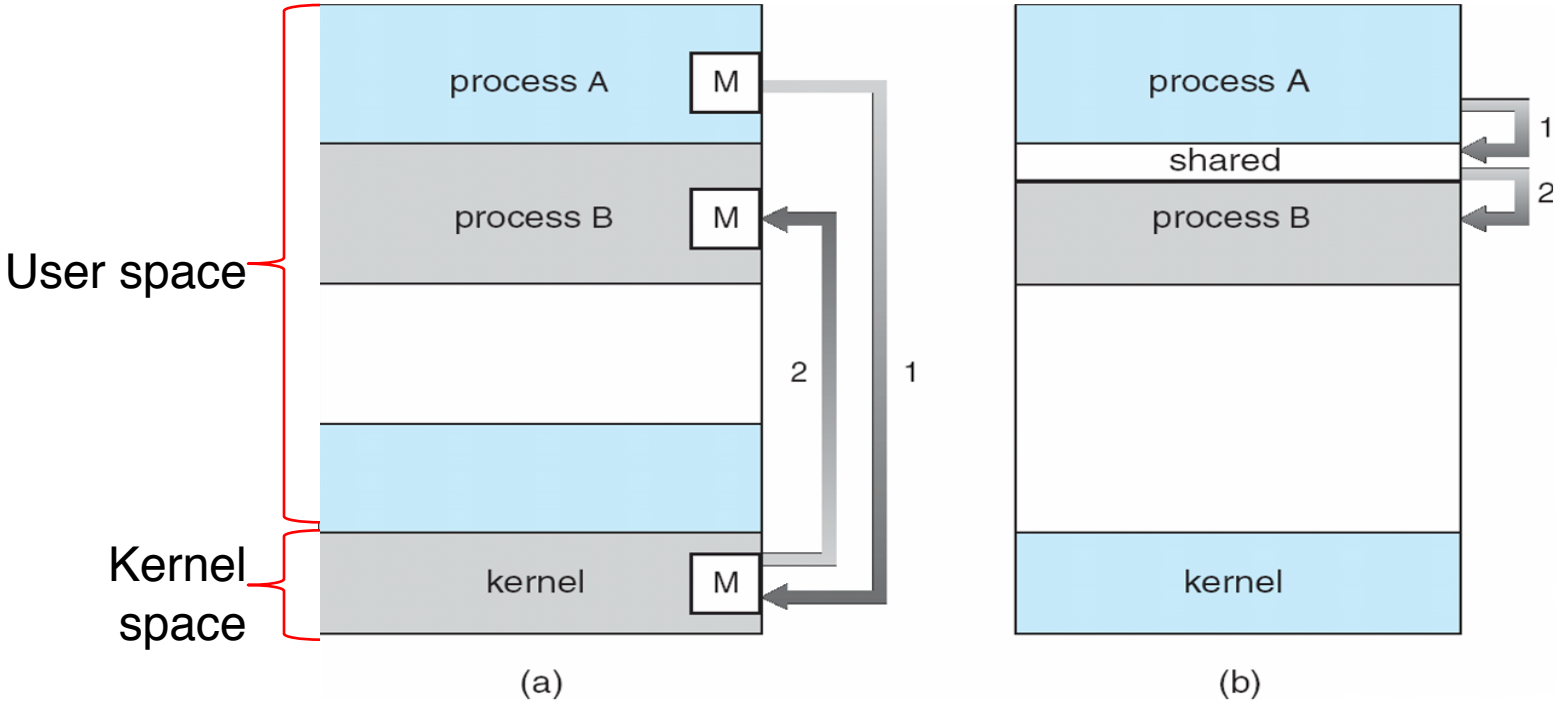
→ False.

Justification: It is used by a parent process to wait for the completion of a child process. Sometimes it is also called “join()”, a reference to the popular fork-join processing model. In this model, a parent process forks several children processes for independent computations, and then combines their results using the join() operation.

1d) Message passing based Inter-Process Communication (IPC) consumes less memory than shared memory based IPC.

→ False.

Justification: It consumes more memory.



2. What are two main differences between the data and stack regions of a process memory?

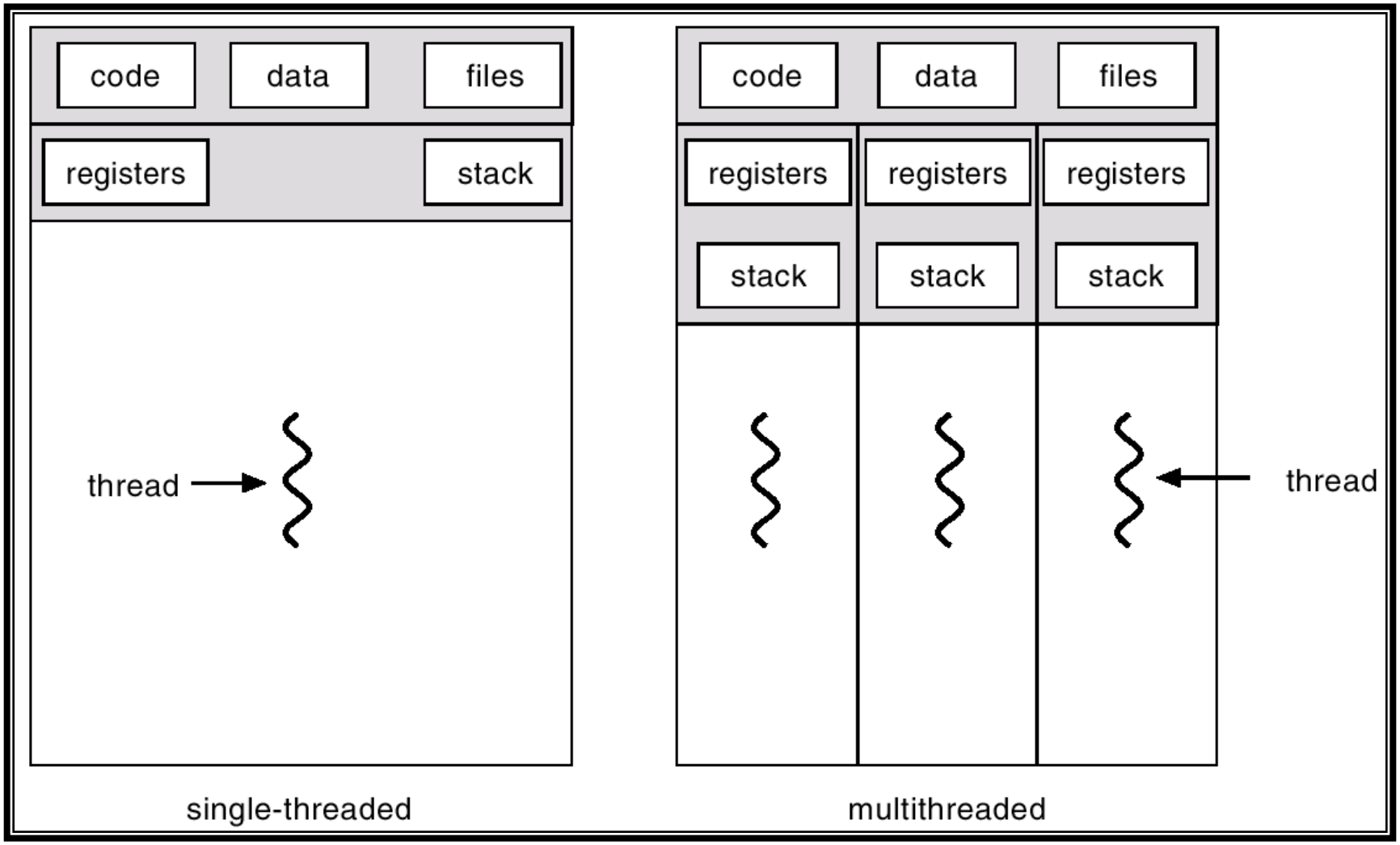
Data vs. Stack Regions of Memory

1. The size of the data region of a process memory is statically fixed, whereas a stack can grow and shrink as the process executes.

2. The data region is used for storing global parameters/variables, whereas the stack region is used for local parameters/variables in functions.

3. Explain the difference between a single-threaded and a multi-threaded process.

Single- vs. Multi-threaded Process



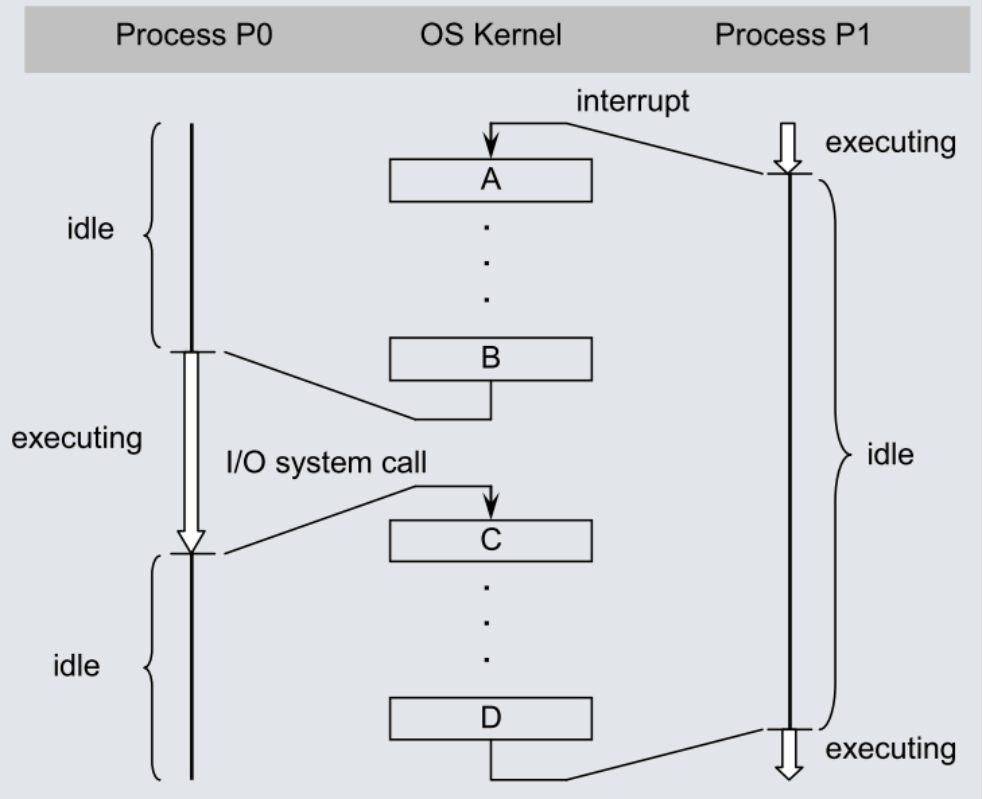
Single- vs. Multi-threaded Process

- Threads in a process share code, data and heap regions of memory, whereas stack space is unique to each thread. Also, each thread has its own Thread Control Block (TCB), similar to a PCB.
- In a single-threaded process, there is only one thread of execution, and hence it is identical to a process.
- In a multi-threaded process, the individual threads can execute concurrently, thus increasing system throughput; when one thread of a process is blocked ("waiting" state), another thread can continue its execution ("running" state).

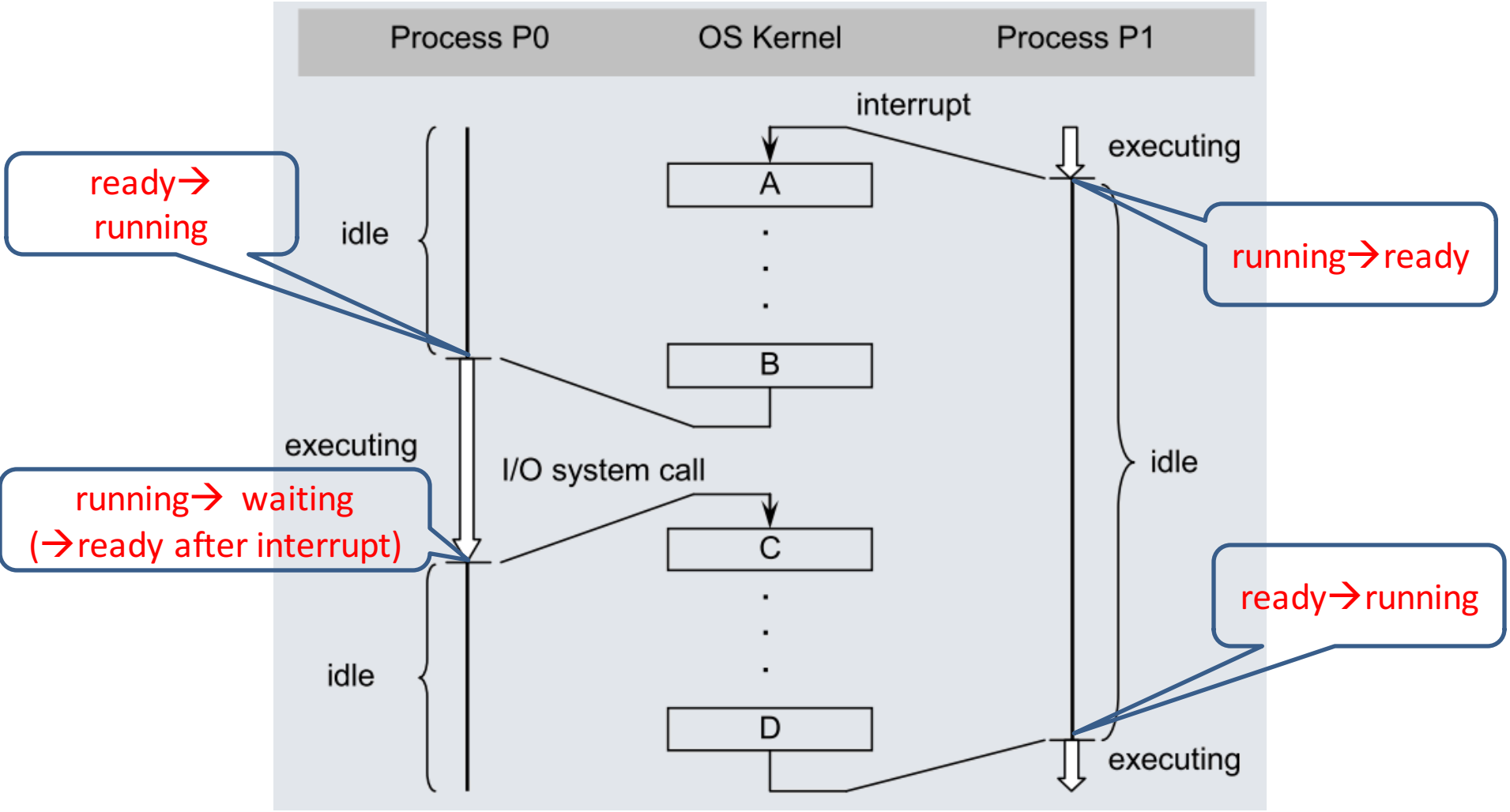
4. The figure below shows the execution of processes P0 and P1 in a multiprogramming system.

a) Identify state transitions of each process.

b) Describe operations A, B, C and D performed by the operating system kernel.



State Transitions



Operations A, B, C and D