Lecture 4: Processes

Learning Goals

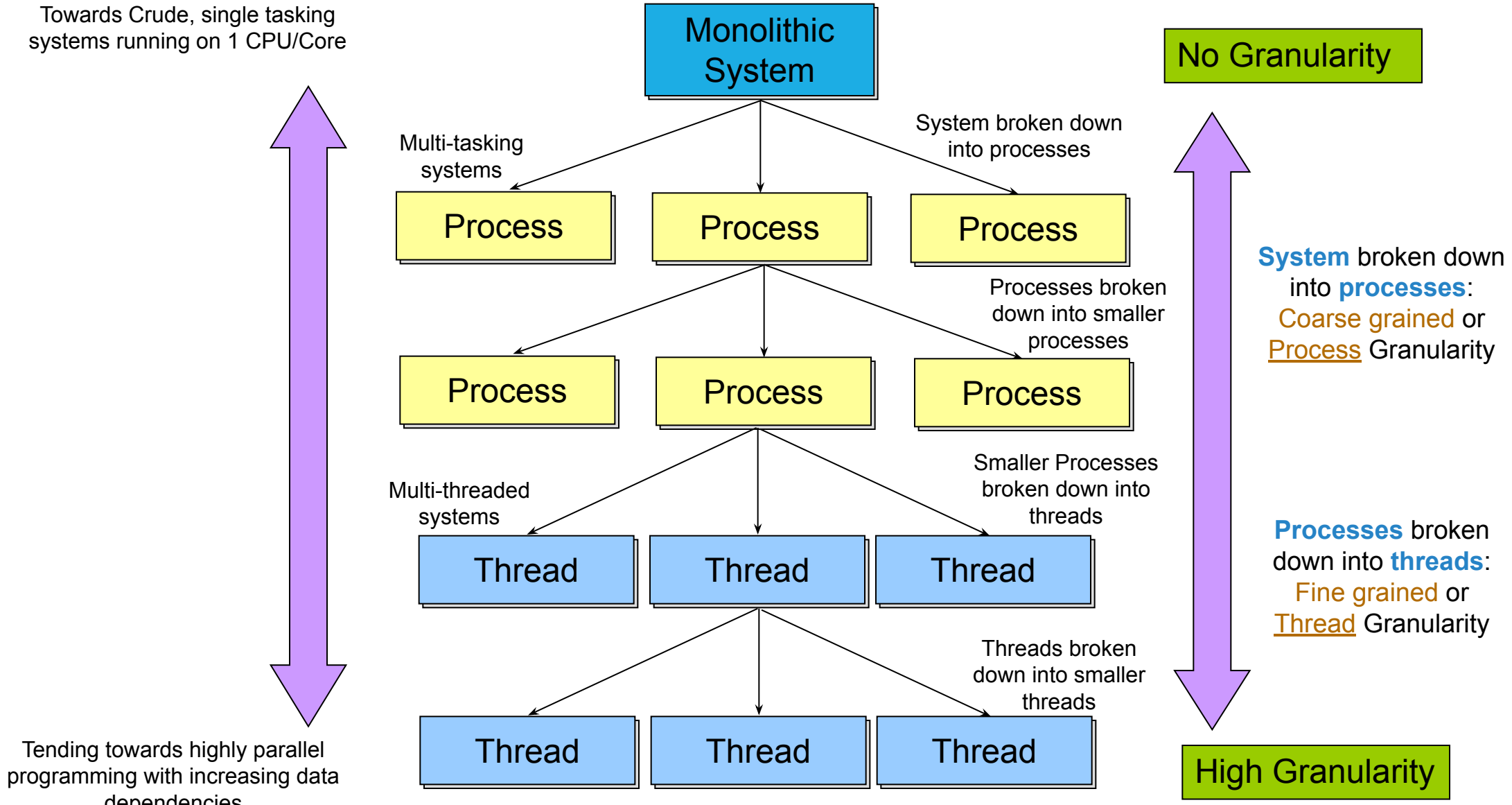
- Describe what a **process** is in your own words
- Describe the role of the Operating System **kernel** in process creation
- Describe the **differences** between processes and threads, and list some **advantages/disadvantages** of each

Process

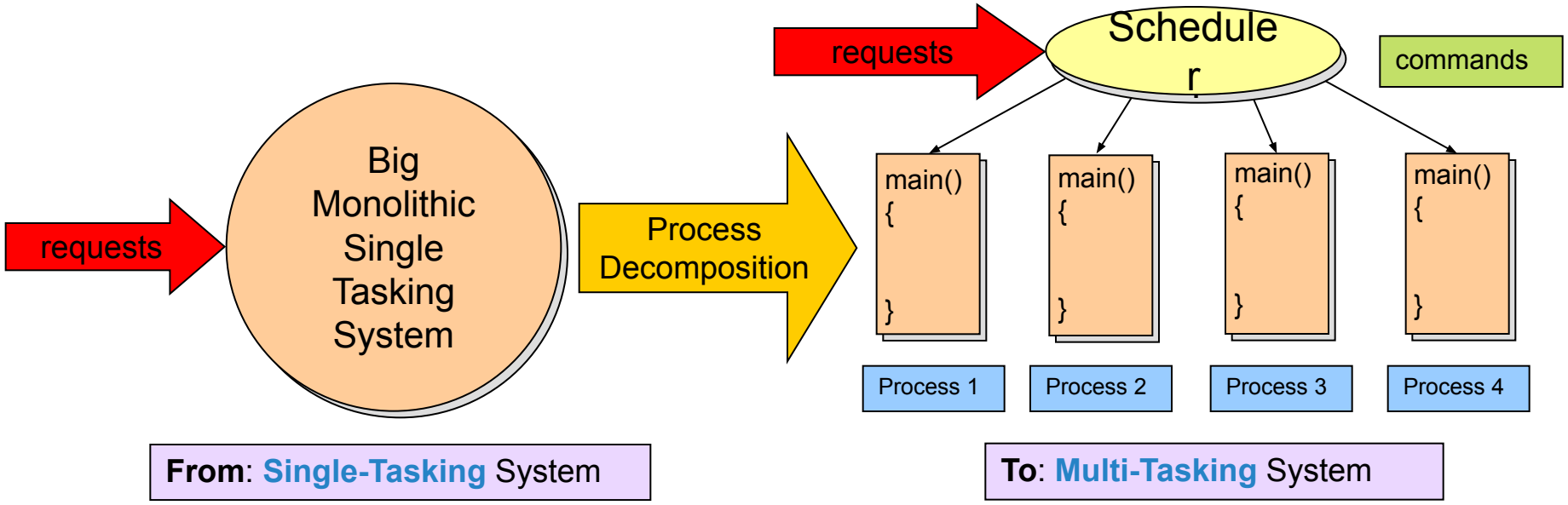
A complete program, consisting of one or more threads of execution, and an **environment**.

- Executable machine code
- Allocated memory block of virtually addressable memory (stack and heap)
- Operating-system-specific descriptors of resources (e.g. file descriptors, handles, data sources/sinks)
- Security attributes (e.g. process owner and permissions)

Application/Process/Thread Decomposition



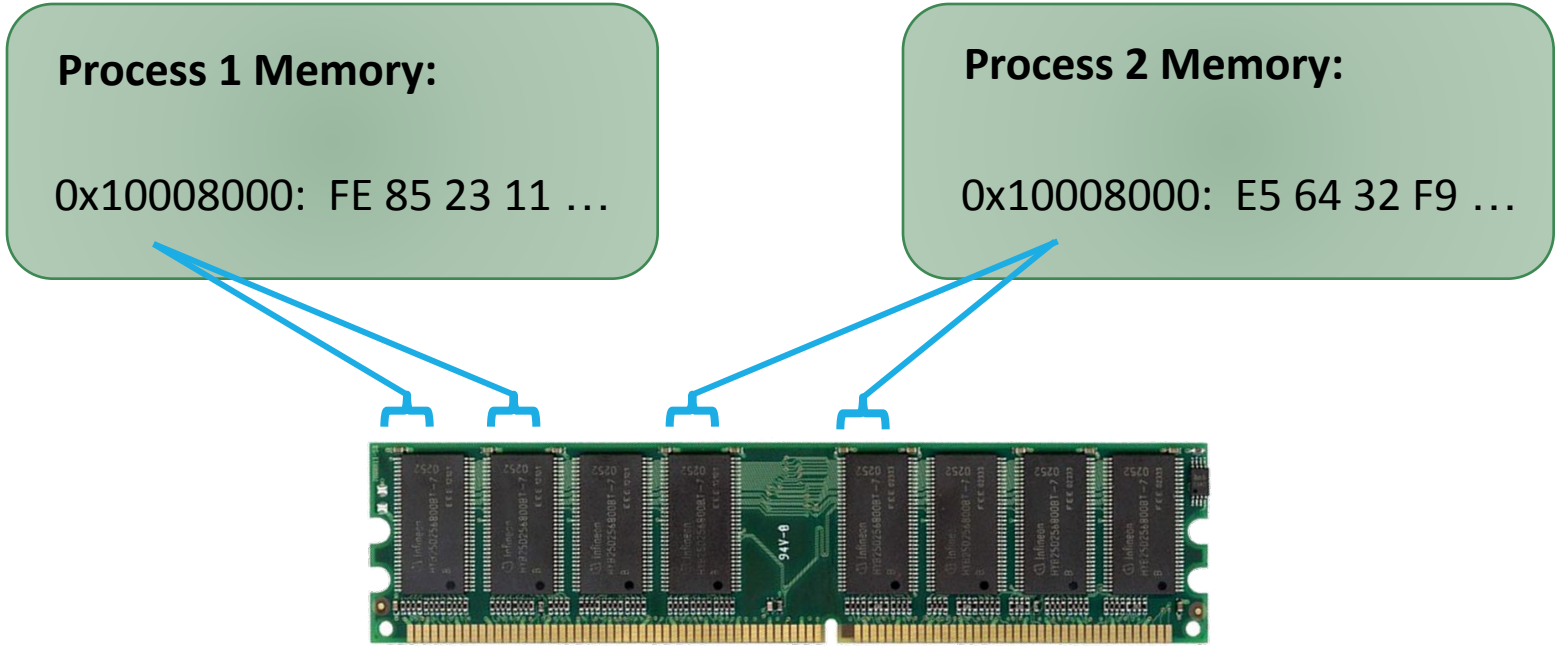
Process Decomposition



- What **guidelines** should the programmer consider when decomposing a system?
- How many processes and threads should we **create**?

Challenges with Processes

Unlike threads within a single process, separate processes **do not** share **address space**.



Challenges with Processes

Unlike threads within a single process, separate processes **do not** share **address space**.

- Multi-tasking operating system kernel
- Support for process creation
- Mechanisms for **inter-process communication**, **inter-process synchronization**

Processes vs Threads

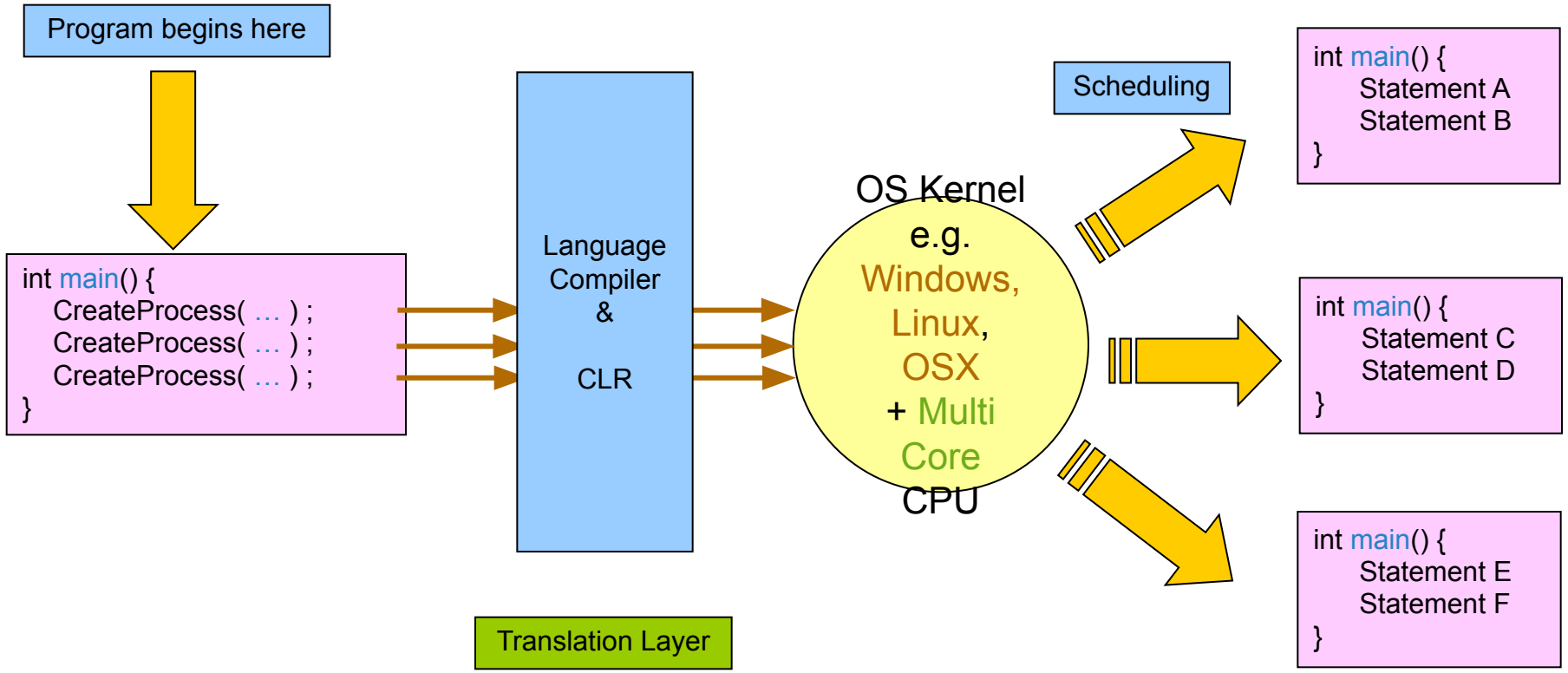
Advantages of Processes:

- If one process fails, others continue
- New process instances can be started manually, or outside of a main program
- Processes can be run on separate machines

Disadvantages of Processes:

- Less efficient than threads due to additional overhead (more memory, state info)
- Communication/synchronization happens across process boundaries

Library Abstractions



Creating a Process (Windows OS)

```
using System;
using System.Diagnostics;

namespace MyProcess
{
    class MyProcess
    {
        public static void Main()
        {
           using (Process myProcess = new Process())
                {
                    myProcess.StartInfo.UseShellExecute = false;
                    myProcess.StartInfo.FileName = "C:\\HelloWorld.exe";
                    myProcess.StartInfo.CreateNoWindow = true;
                    myProcess.Start();
                }
            }
        }
    }
}
```

Creating a Process (Mac & Linux OS)

```
using System;
using System.Diagnostics;

namespace MyProcess
{
    class MyProcess
    {
        public static void Main()
        {
            using (Process myProcess = new Process())
            {
                myProcess.StartInfo.UseShellExecute = false;
                myProcess.StartInfo.FileName = "/Users/ali/github/csharp/Lectures/L4/hello_world.sh";
                myProcess.StartInfo.CreateNoWindow = true;
                myProcess.Start();
            }
        }
    }
}
```