|  |  |  |
| --- | --- | --- |
| **ACTIES OP HET SCHERM** | **VOICE-OVER** | **DUUR** |
|  | Modern CPUs have many cores, each capable of executing an independent instruction thread. However, your program needs to be designed to take advantage of that potential. One of the easiest ways to accomplish this for scientific computing is OpenMP. You annotate with directives, and the compiler will generate the low-level code to spawn threads and divide the computations over them. This is a very extensive topic, and PRACE has some excellent courses dealing with OpenMP, so this is just to give you the flavor. |  |
| 1. Show random\_mod\_01.f90 | 1. Show allocatables |  |
| 1. Show main\_01.f90 | 1. Show error handling for command line arguments |  |
| 1. Compile and run | 1. Discuss runtime error |  |
| 1. Show random\_mod\_02.f90 | 1. Discuss stat argument |  |
| 1. Compile and run | 1. Discuss error message |  |
| 1. Show random\_mod\_03.f90 | 1. Discuss optional ierr |  |
| 1. Show main\_02.f90 | 1. Discuss handling error in main context |  |
| 1. Show main\_03.f90 | 1. Discuss negative arguments |  |
| 1. Compile and run |  |  |
|  |  |  |
| **TOTALE DUUR** | | *Maak je screencast niet langer dan ca. 6 minuten.* |