

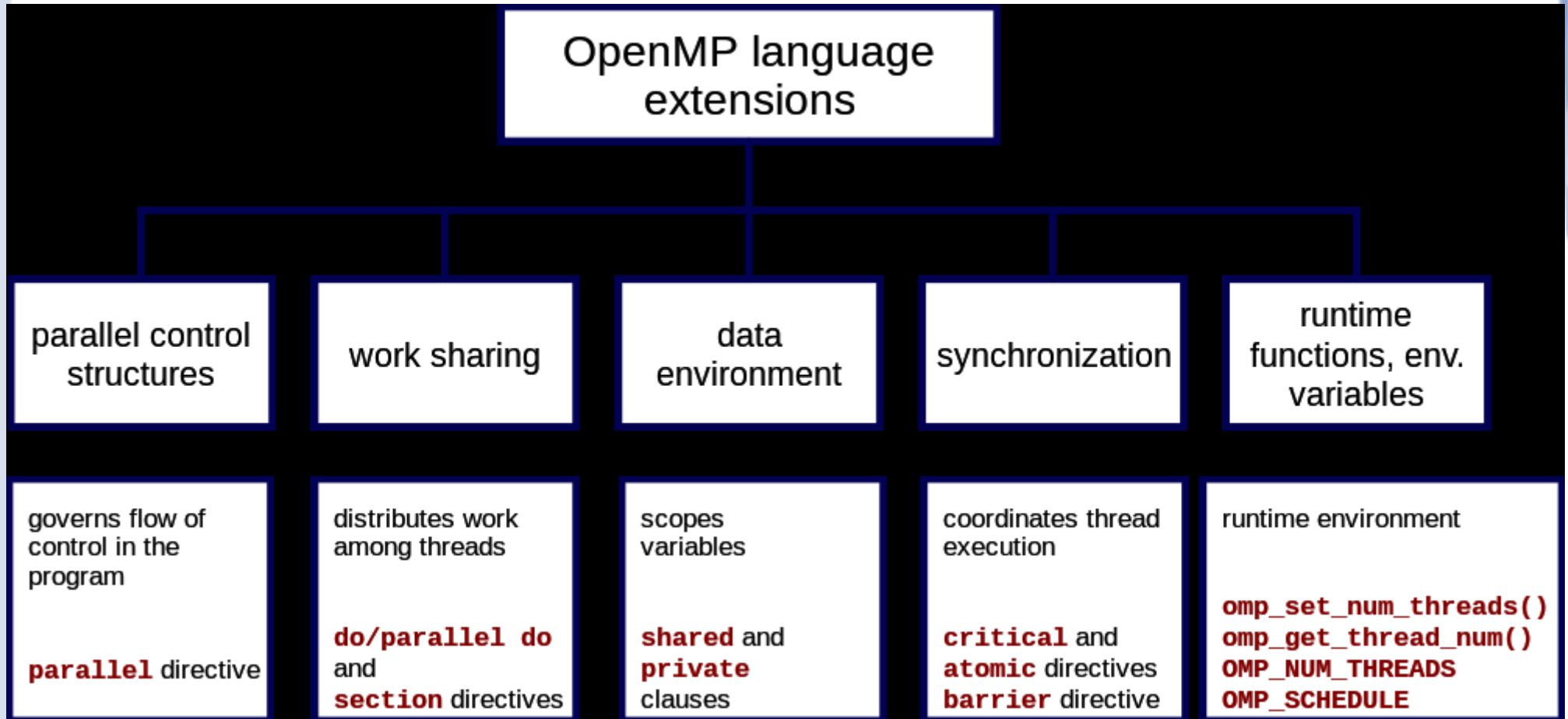
# OpenMP (a.k.a multithreading)

Not to be confused with OpenMPI

# OpenMP

- Threading made simple and readable
- Supported by most popular compilers and also coming to LLVM/CLANG 3.3 (it seems though it was promised for 3.2)

# OpenMP features (borrowed from Wikipedia)



# Some easy unsafe avoidance

```
Complex Matrix phasor(nx, ny);
//You have already defined nx, ny, startRow and endRow
Bool delphase;
Complex * phasorstor=phasor.getStorage(delphase);
//using a private variable Vector<Double> length nx
Bool del;
const Double * visfreqstor=interpVisFreq_p.getStorage(del);
Int irow;
#pragma omp parallel default(none) private(irow) firstprivate(visfreqstor,phasorstor) shared(startRow, endRow, nx)
{
#pragma omp for
  for (irow=startRow; irow<=endRow;irow++){
    doSomeMath(visfreqstor, nx, phasorstor, irow);
  }
}
//end pragma parallel
//Now make sure that the contiguous array be pput back into the object in case getStorage did a copy
phasor.putStorage(phasorstor, delphase);
```

**Check casaindex Notes on experiences using  
openMP in CASA**