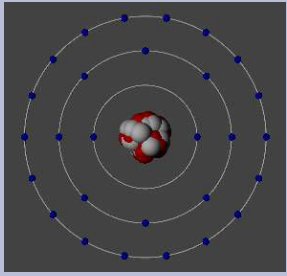


# Паралелно програмиране с MPI



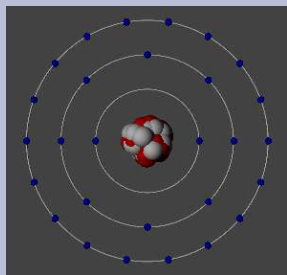
## Основи на MPI



# Манипулатори



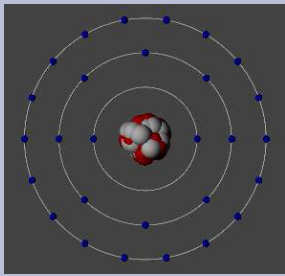
- Базов обект в MPI
- MPI поддържа набор от вътрешни структури, достъпът до които се осъществява само чрез манипулатори
- Прости типове - подлежат на копиране чрез присвояване
- Манипулатори за:
  - комуникатори
  - типове данни
  - паралелен в/и при MPI 2.0



## Типове данни



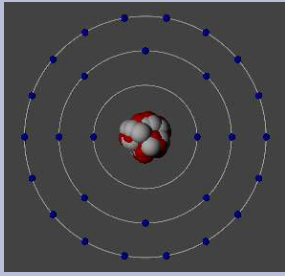
- Съобщенията в MPI са строго типизирани
- Възможностите за изпълнение на MPI програми върху хетерогенни архитектури налагат ограничения върху типовете данни
- MPI има предефинирани типове за всички прости типове на C и Fortran
- Възможност за създаване на производни типове - матрици, подматрици, отрязъци и др.



# Типове данни от C



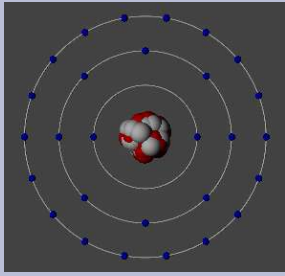
<i>Тип данни в MPI</i>	<i>Тип данни в C</i>
MPI_CHAR	signed char
MPI_SHORT	signed short
MPI_INT	signed int
MPI_LONG	signed long int
MPI_UNSIGNED_CHAR	unsigned char
MPI_UNSIGNED_SHORT	unsigned short
MPI_UNSIGNED	unsigned int
MPI_UNSIGNED_LONG	unsigned long int
MPI_FLOAT	float
MPI_DOUBLE	double
MPI_LONG_DOUBLE	long double
MPI_BYTE	
MPI_PACKED	



# Типове данни от Fortran



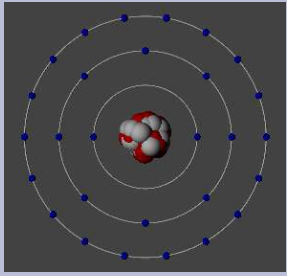
<i>Тип данни в MPI</i>	<i>Тип данни във Fortran</i>
MPI_INTEGER	INTEGER
MPI_REAL	REAL
MPI_DOUBLE_PRECISION	DOUBLE PRECISION
MPI_COMPLEX	COMPLEX
MPI_LOGICAL	LOGICAL
MPI_CHARACTER	CHARACTER(1)
MPI_BYTE	
MPI_PACKED	



# Конвенции в MPI



- Конвенция на именуване в MPI:
  - C - `MPI_First_letter_upper_other_lower_case()`
  - Fortran 77 - `MPI_ALL_LETTERS_UPPER()`
  - Fortran 90+ - `mpi_case_does_not_matter()`
  - константи - `MPI_ALL_LETTERS_UPPER`
- Всички MPI извиквания са:
  - функции в C, които връщат `int` код за грешка
  - подпрограми във Fortran, с допълнителен последен изходен аргумент от тип `INTEGER`
  - изключение: `MPI_Wtime`, `MPI_Wtick` връщат `double` (C) или `DOUBLE PRECISION` (Fortran) и не вземат аргументи



# Инициализация и деинициализация на MPI



- MPI библиотеката следва да се инициализира с `MPI_Init`
  - C: `int MPI_Init (int* pargc, char **pargv);`
  - Fortran: `SUBROUTINE MPI_INIT(ierror)`
- MPI трябва да бъде спряна с `MPI_Finalize`
  - C: `int MPI_Finalize (void);`
  - Fortran: `SUBROUTINE MPI_FINALIZE()`
  - след извикването му никакви MPI извиквания повече!
  - изчиства изходните буфери
  - приключва комуникациите