Excersize 2, Summer 2017

Roman Čunť

Compiling on unix/linux, cross-linking Fortran and C

- make, make -f <makefile>
- Makefile structure, variables, rules
- 2 stages of building an executable: compiling, linking

compiling creates object files, PIC machine code

linking is called when "program" (Fortran) or "int main()" are encountered. Selected important differences between C and Fortran

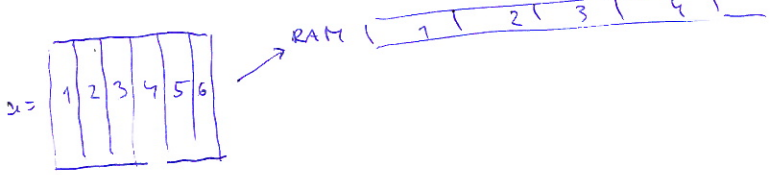
Fortran

1.) Arguments always passed by the reference

```
real(8) function func(a)
    real(8):: a
end function
```

2.) Memory alignment of multi-indexed arrays:

```
real(8):: a(N,N)
```



3.) Fortran compilers appends an underscore to the compiled function when they generate an object file

C

1.) Arguments of function passed by value or by reference:

a.) value: `double func(double a)`

b.) reference: `double func(double *a)`

2.) `double a[N][N]`

3.) C does not modify the names of functions into the object files

C++ mangles though! (due to overloading)

object file generated by

Fortran:	C:
func1_	func1
func2_	func2

So when cross-linking you encounter 2 situations:

1.) Fortran functions are used in C-code (our situation)

a.) Fortran functions are called as ~~b = func(a)~~ b = func_(&a)

b.) they are declared as "func_(double *)"

2.) C-functions are used in Fortran-code

a C-functions must be written as "double func_(double *a)"