

PH502: Scientific Programming Concepts

Irish Centre for High End Computing (ICHEC)

September 23, 2020

- This lecture covers pointers.
- If you have not worked with any compiled languages before, pointers will be a new experience.
- They are essential in C programming. FORTRAN has pointers but they generally not needed.

- A pointer is something that points to an address in memory.
- The pointer itself is a variable and is stored in a different location in memory.
- So the *r – value* of a pointer is a memory address or an *l – value*.
- The amount of memory that a pointer needs is depend on the architecture but for 64-bit machines it will be an 8-byte integer.
- Pointers also need a type, like int float etc.
- This may seem strange but as we shall see later it defines what happens when we perform pointer arithmetic.

- Pointers are variables that point to a location in memory.
- Here *px* is a pointer to an integer variable. *px* is the *l* – value of the variable and **px* the *r* – value.

```
int *px, x;  
// Set l-value to that of variable x, r-value set to 10  
x = 10; px = &x;  
// Set r-value to 1  
*px = 1;  
// Add one to r-value  
(*px)++;  
// ++ is a higher precedence than *  
// Print l-value and r-value  
printf(" l and r values %p, %d\n",px,*px);  
  
// Equivalently  
x= 2;  
printf(" l and r values %p, %d\n",&x,x);
```

- Pointers are not as necessary in FORTRAN but we include them for completeness.

```
integer (kind=4), pointer :: px
integer (kind=4), target :: x

x = 10; px => x;
px = px + 1      ! x = 11
write(6,*) ' x and address ', x, loc(x)
```

- A FORTRAN pointer can only be associated with the same type of variable with a “TARGET” attribute.
- The *l-value* cannot be accessed so there is no equivalent of $px++$ in FORTRAN, *i.e.* pointer arithmetic is not permitted.