

# PH502: Scientific Programming Concepts

Irish Centre for High End Computing (ICHEC)

September 23, 2020

- Finally there are two useful utilities, that are part of the standard libraries.
- These are random number generator.
- And timing utilities.

- `int rand(void)` : returns a pseudo-random number between 0 and `RAND_MAX`(at least 32767).

```
//Generates a number between 0 and 1  
r1=((float) rand() / (RAND_MAX));  
//Generates a number between min and max  
r2=((float) rand() % (max+1-min))+min;
```

- `void srand(unsigned int seed)` : sets the random number generation seed. Used either to make generation more random (use current time) or reproducible (use a set value).
  - ▶ Prevents random numbers from being the same every time the program is executed.

```
int a;  
  
srand(time(0));  
a=rand();
```

- ▶ If seed is set to 1, the generator is reinitialized to its initial value.

- In FORTRAN there are two subroutines, one sets the seed and the other gets a REAL vector of pseudo random numbers.
- If the seed is the same between different executions of the program the set of random numbers is the same.

```
real (kind=4) :: rvec(1000)
call random_seed
call random_number(rvec)
```

- C represents time in two ways:
  - ▶ the number of seconds elapsed since midnight on January 1, 1970. This is stored as a `time_t` which is itself defined as a long integer.
  - ▶ the struct `tm` structure breaking down a point in time into its components: year, month, day...
- Here is the definition of the struct `tm` type:

```
struct tm {  
    int tm_sec;      /* seconds after the minute - [0,59] */  
    int tm_min;      /* minutes after the hour - [0,59] */  
    int tm_hour;      /* hours since midnight - [0,23] */  
    int tm_mday;      /* day of the month - [1,31] */  
    int tm_mon;       /* months since January - [0,11] */  
    int tm_year;      /* years since 1900 */  
    int tm_wday;      /* days since Sunday - [0,6] */  
    int tm_yday;      /* days since January 1 - [0,365] */  
    int tm_isdst;     /* daylight savings time flag */  
};
```

- `time_t time(time_t *timeptr)` returns the current time, and also stores it at the address provided.

```
time_t t;  
t = time(NULL); // fine  
t = time(0);    // 0 == NULL, so works too  
time(&t);       // valid too  
printf("Seconds since 01/01/1970: %ld\n", t);
```

- `char *ctime(time_t *ptr)` converts to a human-readable string representation.

```
time_t t;  
t = time(0);  
printf("%s\n", ctime(&t));  
// Sample output:  
// Sun Feb 16 15:45:22 2014
```

- `struct tm *localtime(const time_t* tp)` converts the time in seconds to the time structure.
- `char *asctime(const struct tm* tp)` converts a time structure to human-readable string representation.

```
time_t t;  
struct tm *mytime;  
  
t = time(0);  
mytime = localtime( &t );  
printf("%s\n", asctime(mytime));  
// Sun Feb 16 15:45:22 2014
```

- `double difftime(time_t time2, time_t time1)` : returns the difference in seconds between *time2* and *time1*.
- `clock_t clock(void)` : returns the time in internal clock units since the current program began its execution. To convert to seconds divide by the constant `CLOCKS_PER_SEC`.

```
int i;
time_t tst, tend;
clock_t cst, cend;

tst = time(0);
cst = clock();
for(i=0; i<=99999999; i++){
    sqrt(i);
}
tend = time(0);
cend = clock();
printf("Elapsed: %lf\n", difftime(tend, tst));
printf("Elapsed: %ld\n", cend-cst);
```



- There are two subroutines in FORTRAN for time.
  1. The first returns the current date, all arguments are optional.

```
character (len=10) :: date,time,zone  
integer (kind=4) :: values(8)  
call date_and_time(date,time,zone,values)
```

2. The other subroutine gives number of counts, number of counts per second, maximum number of counts.

```
integer (kind=4) :: cnt1,cnt2,rcnt,maxcnt  
  
call system_clock(cnt1,rcnt,maxcnt)  
! Do something  
call system_clock(cnt2,rcnt,maxcnt)  
  
write(6,*) ' Time taken ',cnt2-cnt1
```

- This week we discussed:
  1. types of algorithms,
  2. elementary data structures,
  3. standard library functions,
  4. reading and writing to file,
  5. random number generation,
  6. timing functions.