# Just random notes

b0th

August 2, 2021

**Runtime**
Runtime describes software/instructions that are executed while your program is running, especially those instructions that you did not write explicitly, but are necessary for the proper execution of your code.

**KISS** *Keep It simple, stupid*
The KISS principle states that most systems work best if they are kept simple rather than made complicated.

**Container**
Group of namespaces and control groups applied to a process.

**Linux kernel namespace**
Limit what the process sees, here some namespaces

- item
- pid
- net
- mnt
- uts
- ipc
- user

C functions to manage them

- clone()
- unshare()

**Linux kernel cgroup** *Control group*
Limit what the process can use, here some cgroups

- memomry
- CPU
- network
- devices
- pids

**C++ inheritance class**
Single inheritance

```
class Rectangle: public Shape {
    public:
       int getArea() { return (width * height); }
};
```

Multiple inheritance

```
class Rectangle: public Shape1, Shape2, Shape3 {
    public:
       int getArea() { return (width * height); }
};
```

**C++ namespace**
Namespaces allow to group entities like classes, objects and functions under a name. Example of declaration

```
namespace myNamespace
{
  int a = 0;
}
```

Usage

```
std::cout << myNamespace::a << std::endl
```

or

```
using namespace myNamespace;
std::cout << a << std::endl
```

**C++ cout** *character out*

**C++ endl** *end line*

### Makefile special variables

```
all: library.cpp main.cpp
```

```
$@ evaluates to all
$< evaluates to library.cpp
$^ evaluates to library.cpp main.cpp
```

**Web CGI** *Common Gateway Interface*
Set of standards that define how information is exchanged between the web server and a custom script.

**socket**
It's a network connector, it allows communication between two different processes on the same or different machines. To be more precise, it's a way to talk to other computers using standard Unix file descriptors.

```
int socket(int domain, int type, int protocol);
```

domain (socket protocol) examples

**Local communication** AF_UNIX, AF_LOCAL

**IPv4 Internet protocols** AF_INET

**IPv6 Internet protocols** AF_INET6

type (precise persistent connection or not) examples

**Two-way reliable communication** SOCK_DSTREAM

**Connectionless** SOCK_DGRAM