Just random notes

b0th

August 3, 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);
```

0.1 domain (socket protocol) examples

Local communication AF_UNIX, AF_LOCAL

IPv4 Internet protocols AF_INET

IPv6 Internet protocols AF_INET6

0.2 type (precise persistent connection or not) examples

Two-way reliable communication SOCK_DSTREAM

Connectionless SOCK_DGRAM

C++ static method

```
class Rectangle {
    public:
        static int perimeter;
}
...
std::cout << Rectangle::perimeter << std:endl
```

C++ reference vs pointer
Both value are implemented by storing the adress of an object but there are some differences

Table 1: Differences between a reference and a pointer

Main points	**Reference**	**Pointer**
Initialization	Declare and initialize	Declare and initialize pointer at same step or in multiple line
Reassignment	Banned	Allowed
Memory adress	Share the same memory address with the original variable (takes some place in the stack too)	Own memory address and size on stack
NULL value	Banned	Allowed
Indirection (pointer to pointer as example)	Banned	Allowed

C++ protected
It allows derived class to acces base class variables