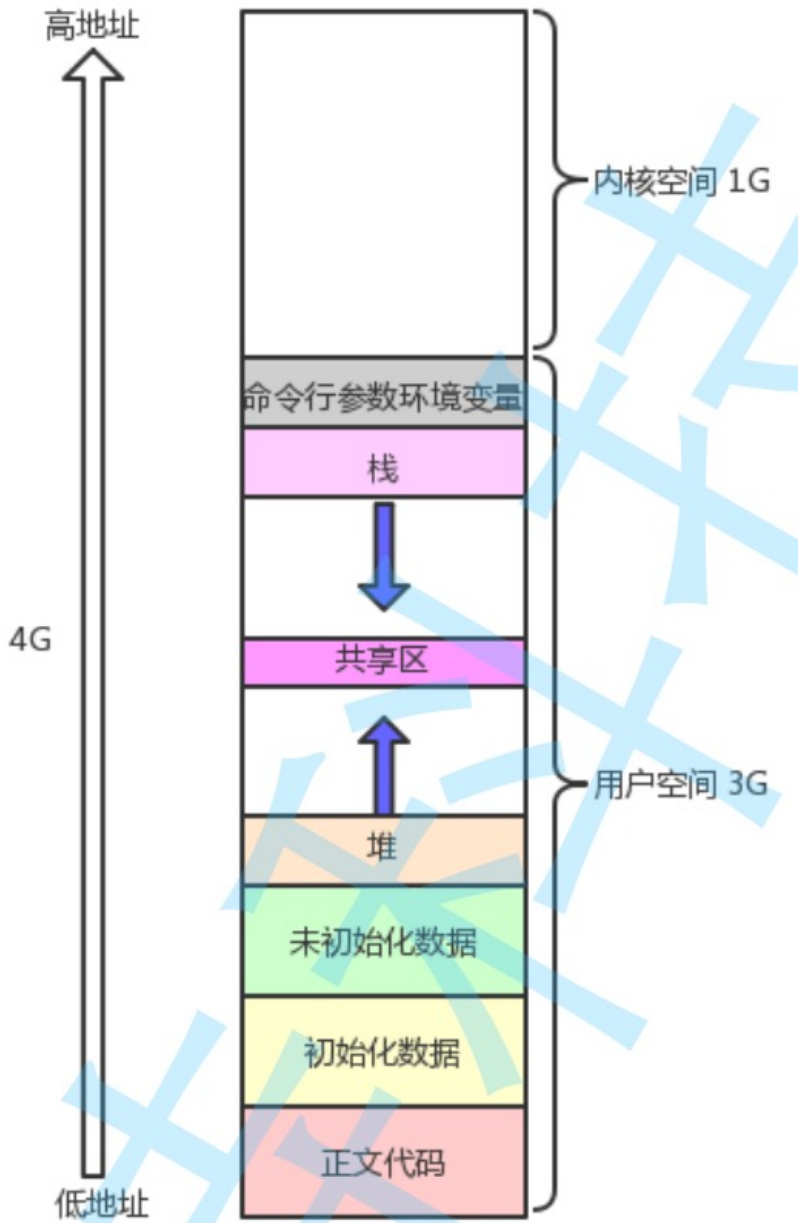
0907 复习+进程地址空间

今天的目标：

1. 1h左右，复习，主线，重点回顾

2. 进程地址空间
    a. 认识概念 – 做实验
    b. 什么是地址空间
    c. 为什么要有地址空间
    d. 地址空间是怎么设计的



这个位置是内存吗？

这个不是内存！

```c
#include<stdio.h>
#include<unistd.h>
int g_val = 100; 定义一个全局变量
int main()
{
    pid_t id = fork();
    if(id == 0)
    {
        int cnt = 0;
        //child
        while(1)
        {
            printf("I am child, pid: %d,ppid: %d, g_val: %d, &g_val: %p\n",
                   getpid(),getppid(),g_val,&g_val);
            sleep(1);
            cnt++;
            if(cnt == 5) 仅在子进程中，改成200
            {
                g_val = 200;//把全局改成200
                printf("child change g_val 100->200 success\n");
            }
        }
    }
    else
    {

        //father
        while(1)
        {
            printf("I am father, pid: %d,ppid: %d, g_val: %d, &g_val: %p\n",
                   getpid(),getppid(),g_val,&g_val);
            sleep(1);
        }
    }
    return 0;
}
```

```
[yufc@learningmechine 0907]$ ls
hello  hello.c  Makefile
[yufc@learningmechine 0907]$ ./hello
I am father, pid: 4937,ppid: 2121, g_val: 100, &g_val: 0x42004c
I am child, pid: 4938,ppid: 4937, g_val: 100, &g_val: 0x42004c
I am child, pid: 4938,ppid: 4937, g_val: 100, &g_val: 0x42004c
I am father, pid: 4937,ppid: 2121, g_val: 100, &g_val: 0x42004c
I am child, pid: 4938,ppid: 4937, g_val: 100, &g_val: 0x42004c
I am father, pid: 4937,ppid: 2121, g_val: 100, &g_val: 0x42004c
I am father, pid: 4937,ppid: 2121, g_val: 100, &g_val: 0x42004c
I am child, pid: 4938,ppid: 4937, g_val: 100, &g_val: 0x42004c
I am father, pid: 4937,ppid: 2121, g_val: 100, &g_val: 0x42004c
I am child, pid: 4938,ppid: 4937, g_val: 100, &g_val: 0x42004c
child change g_val 100->200 success
I am child, pid: 4938,ppid: 4937, g_val: 200, &g_val: 0x42004c
I am father, pid: 4937,ppid: 2121, g_val: 100, &g_val: 0x42004c
I am father, pid: 4937,ppid: 2121, g_val: 100, &g_val: 0x42004c
I am child, pid: 4938,ppid: 4937, g_val: 200, &g_val: 0x42004c
I am child, pid: 4938,ppid: 4937, g_val: 200, &g_val: 0x42004c
I am father, pid: 4937,ppid: 2121, g_val: 100, &g_val: 0x42004c
I am father, pid: 4937,ppid: 2121, g_val: 100, &g_val: 0x42004c
I am child, pid: 4938,ppid: 4937, g_val: 200, &g_val: 0x42004c
çI am child, pid: 4938,ppid: 4937, g_val: 200, &g_val: 0x42004c
I am father, pid: 4937,ppid: 2121, g_val: 100, &g_val: 0x42004c
I am father, pid: 4937,ppid: 2121, g_val: 100, &g_val: 0x42004c
I am child, pid: 4938,ppid: 4937, g_val: 200, &g_val: 0x42004c
^C
[yufc@learningmechine 0907]$
```

我们观察地址，观察变量的值
我们发现

子进程把数据修改完之后
我们发现 – 两个变量的地址还是一样的
这是为什么？一个地址放着两个不同的值？

首先我们可以立马得出的一个结论：

这里的两个地址，存着不同的值

这两个地址不是物理内存上的地址！！！

蛋哥告诉我们：
不是物理地址 – 而是：虚拟地址（线性地址）！

所以，几乎所有的语言，如果他有“地址”的概念
这个地址一定不是物理地址，而是虚拟地址

其实我们平时访问的内存

其实都是虚拟地址

而虚拟地址 – 是物理内存的映射