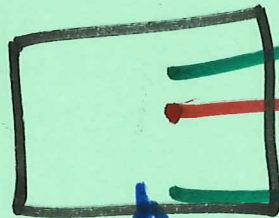


Review

1. C programming
2. Data rep.
3. Assembly program.
- 4 Cache memories.

Physical Addressing.

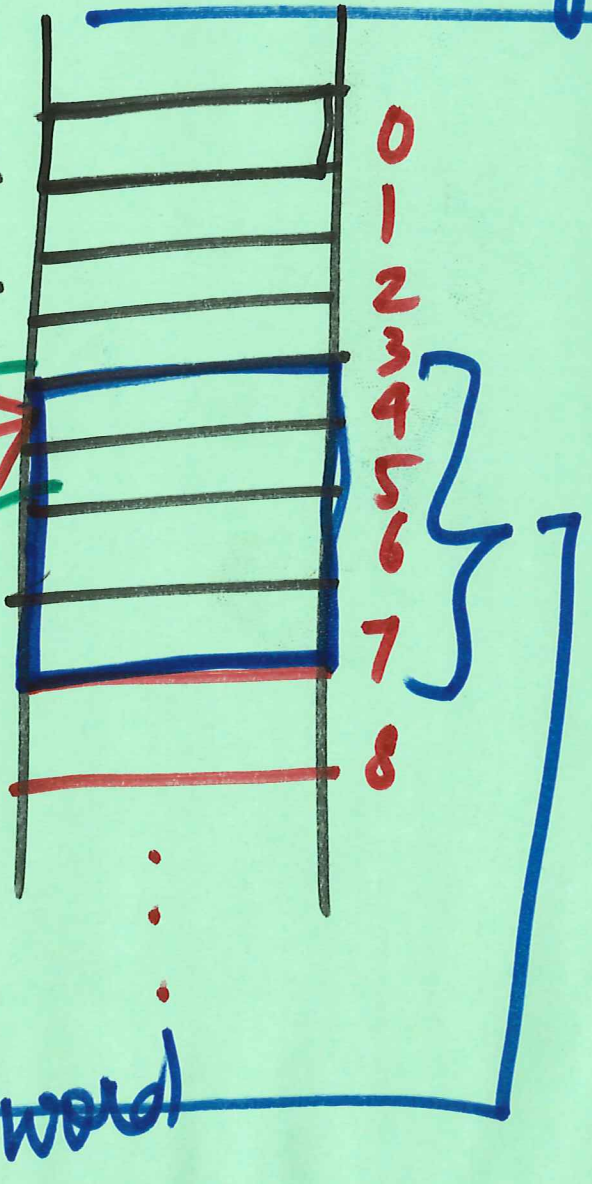
CPU



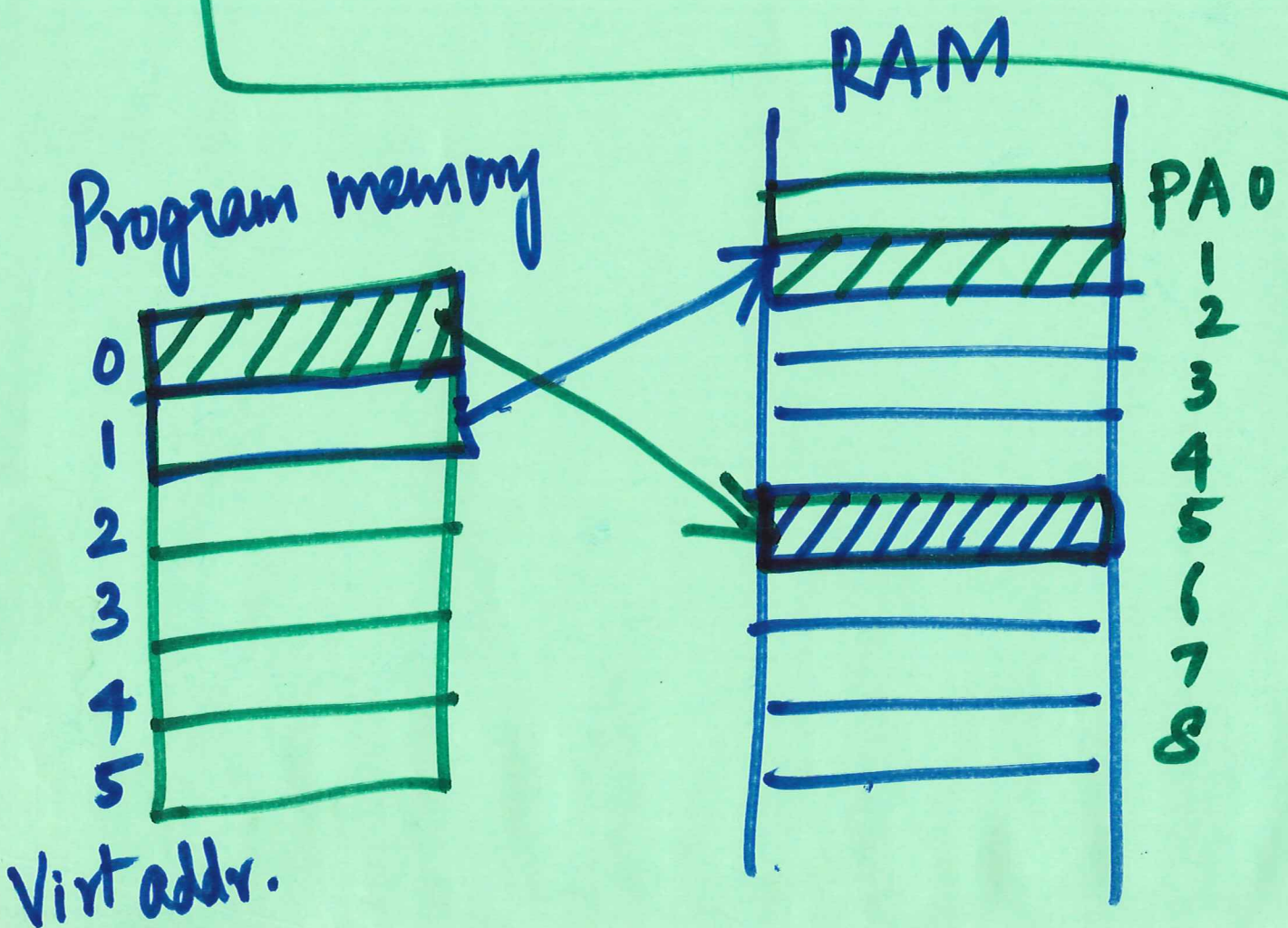
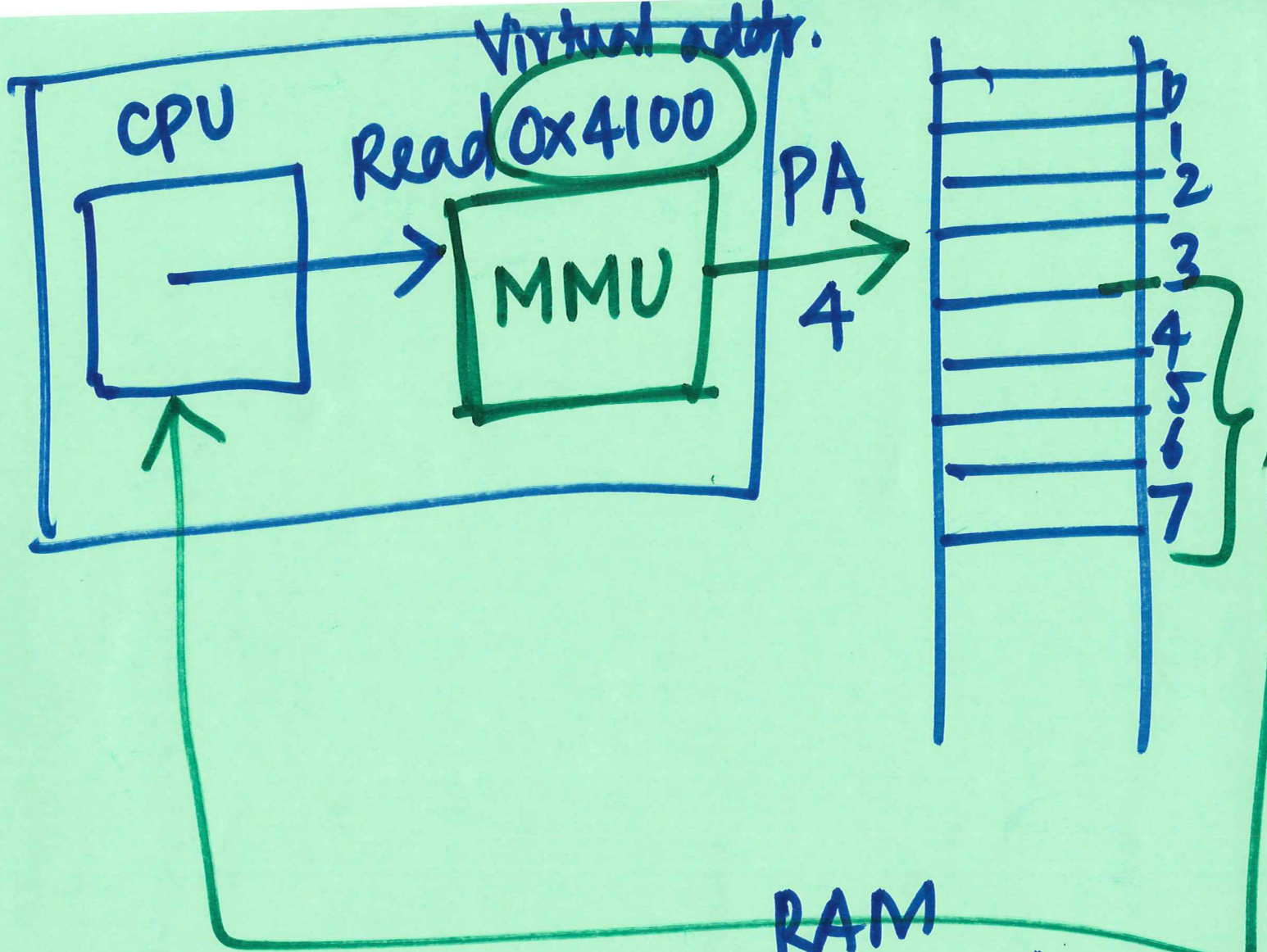
Memory bus

Read word at addr 4.

Main Memory



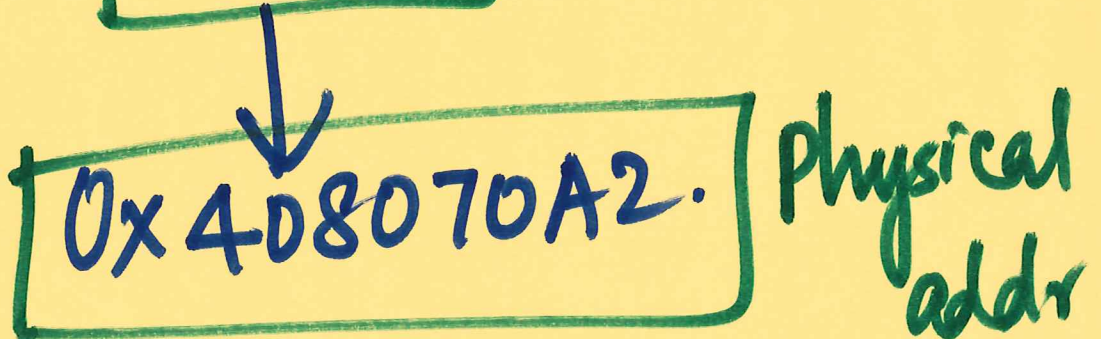
data word

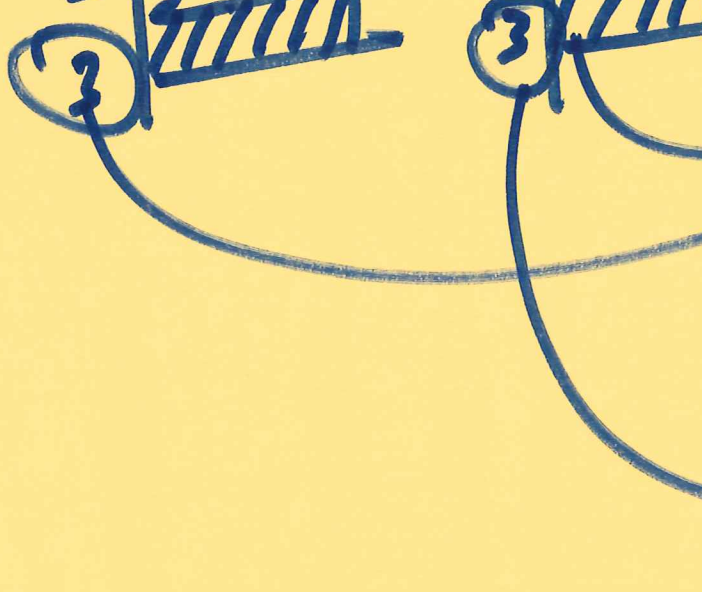
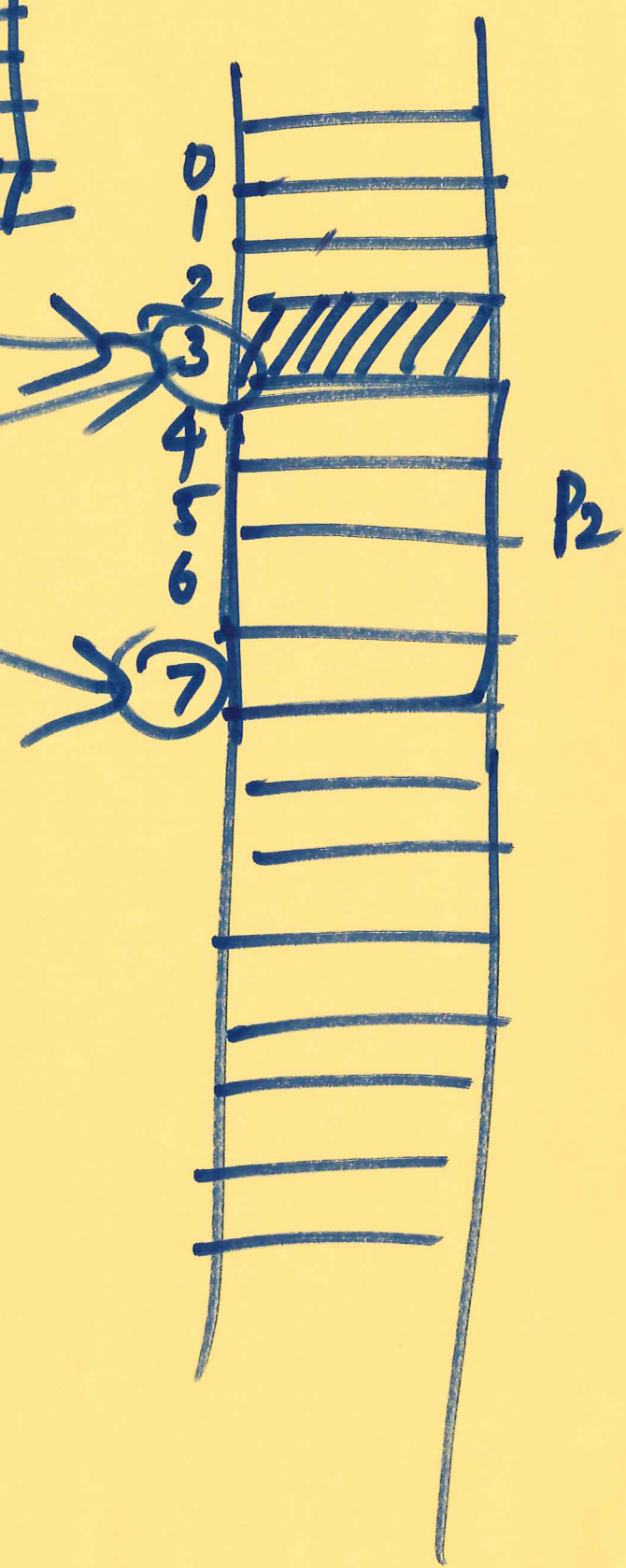
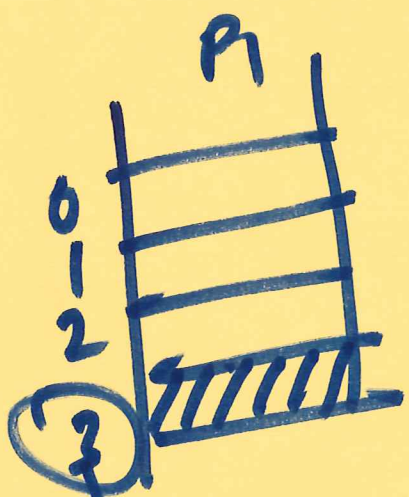


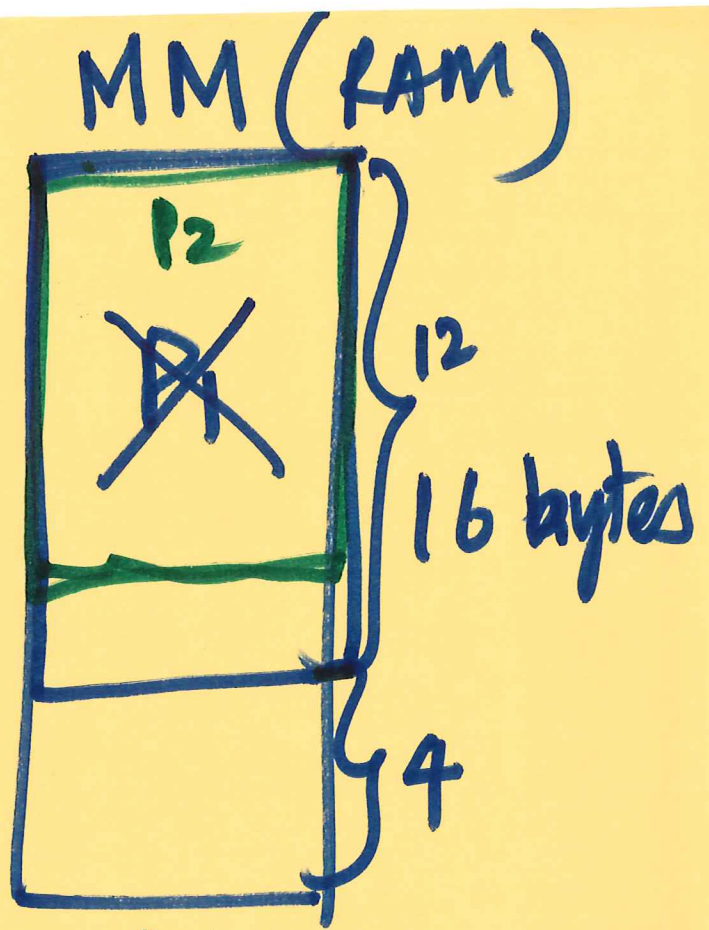
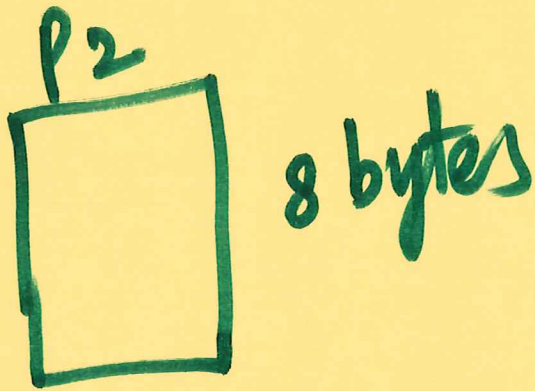
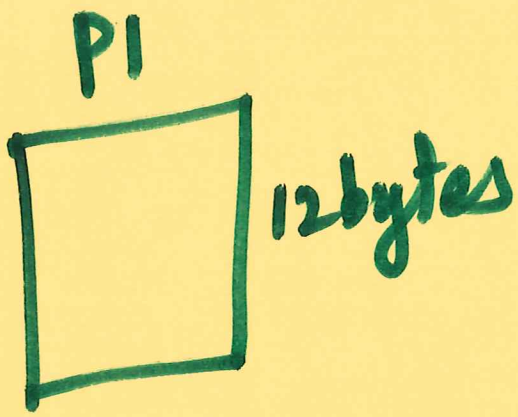
Virtual Memory

MOV 0x40900001, %eax

Virtual addr.

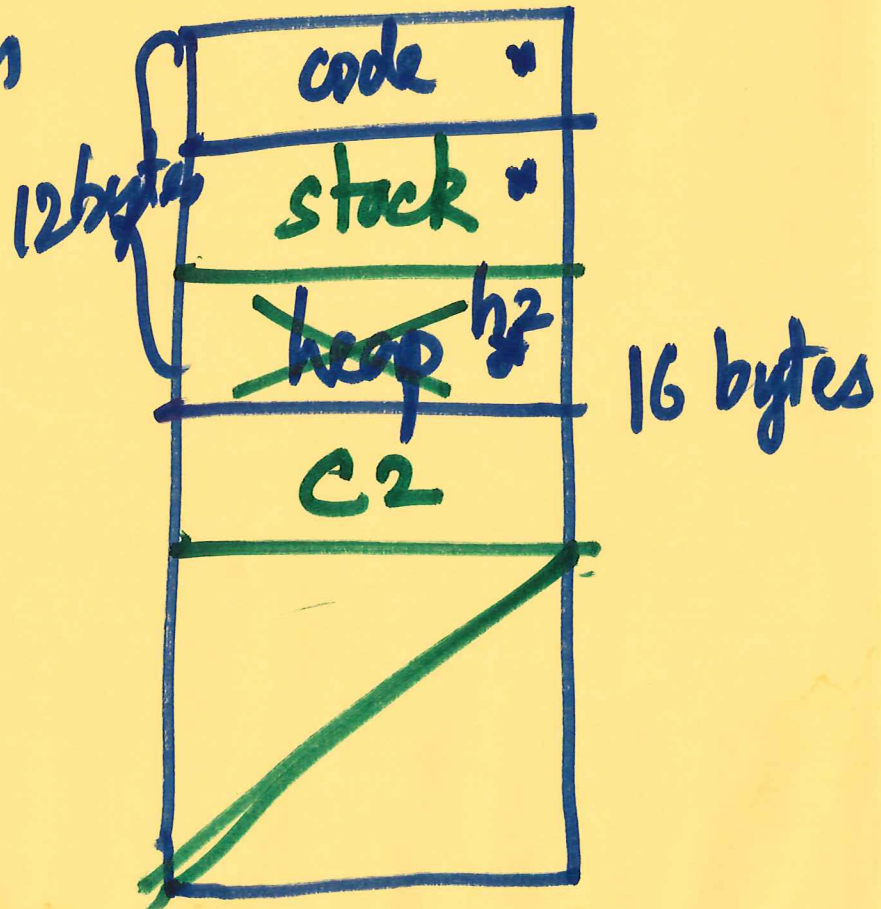
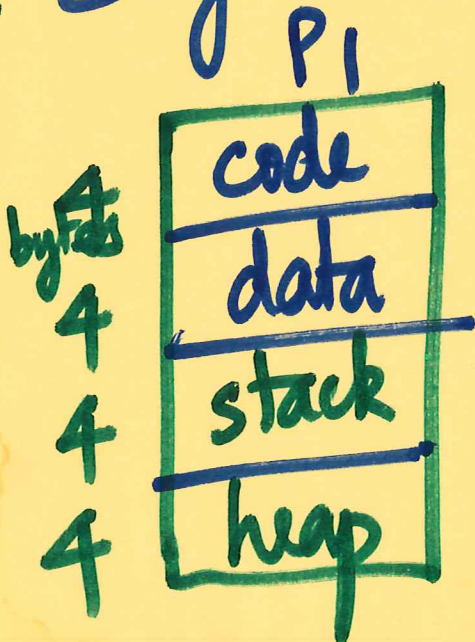


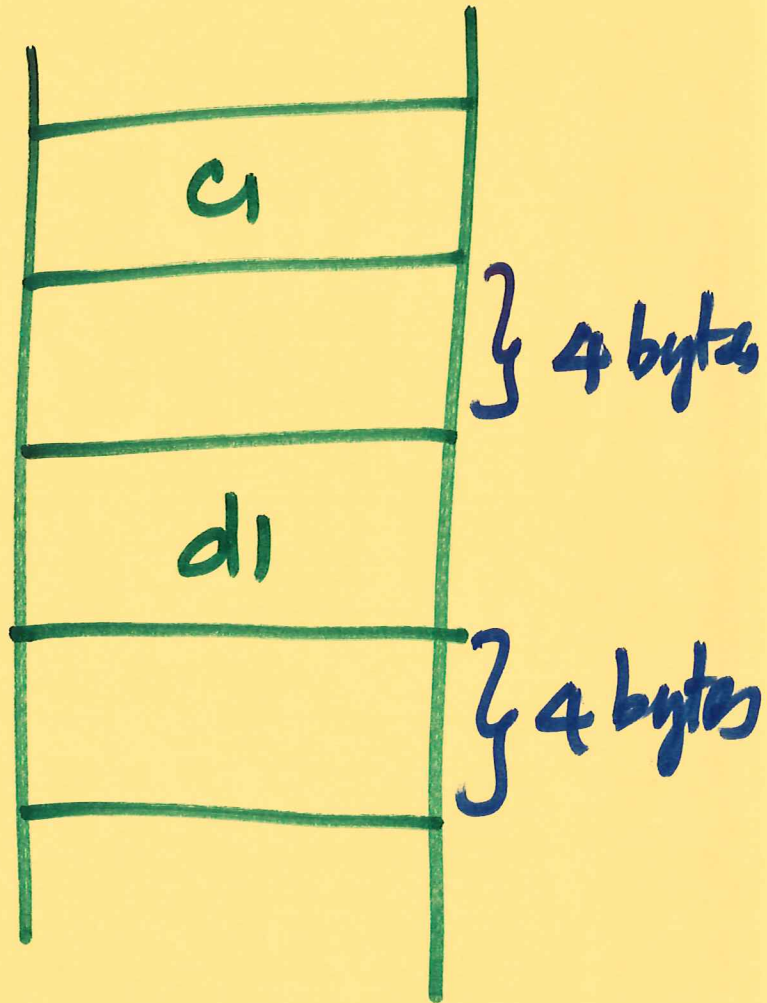
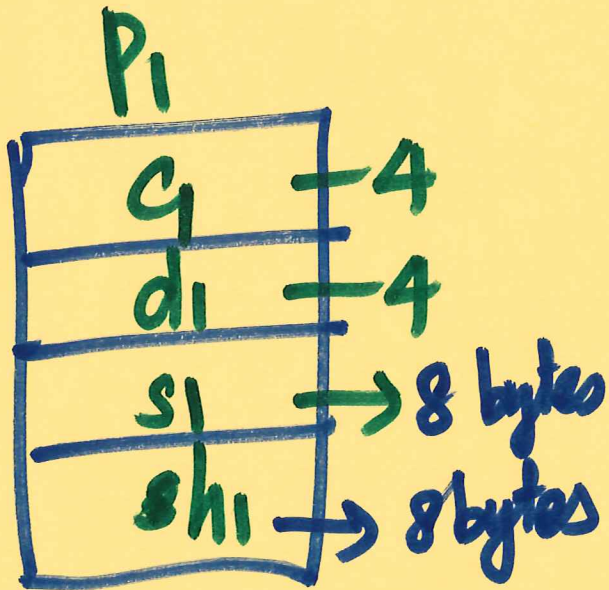
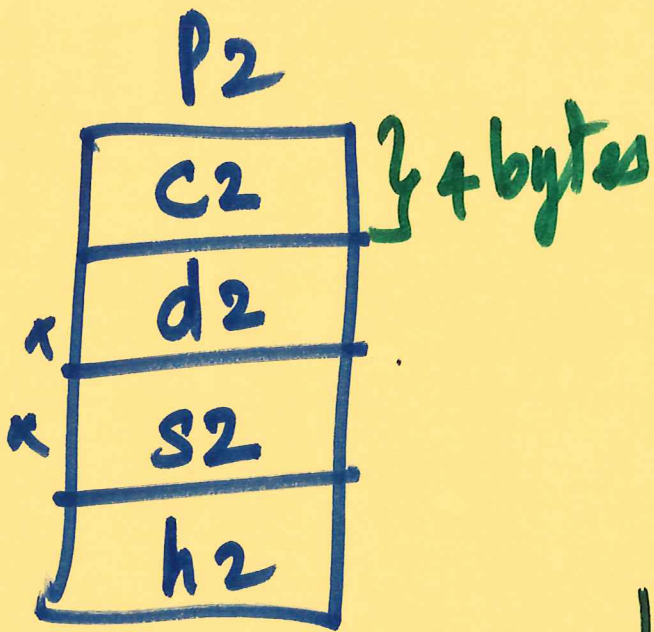


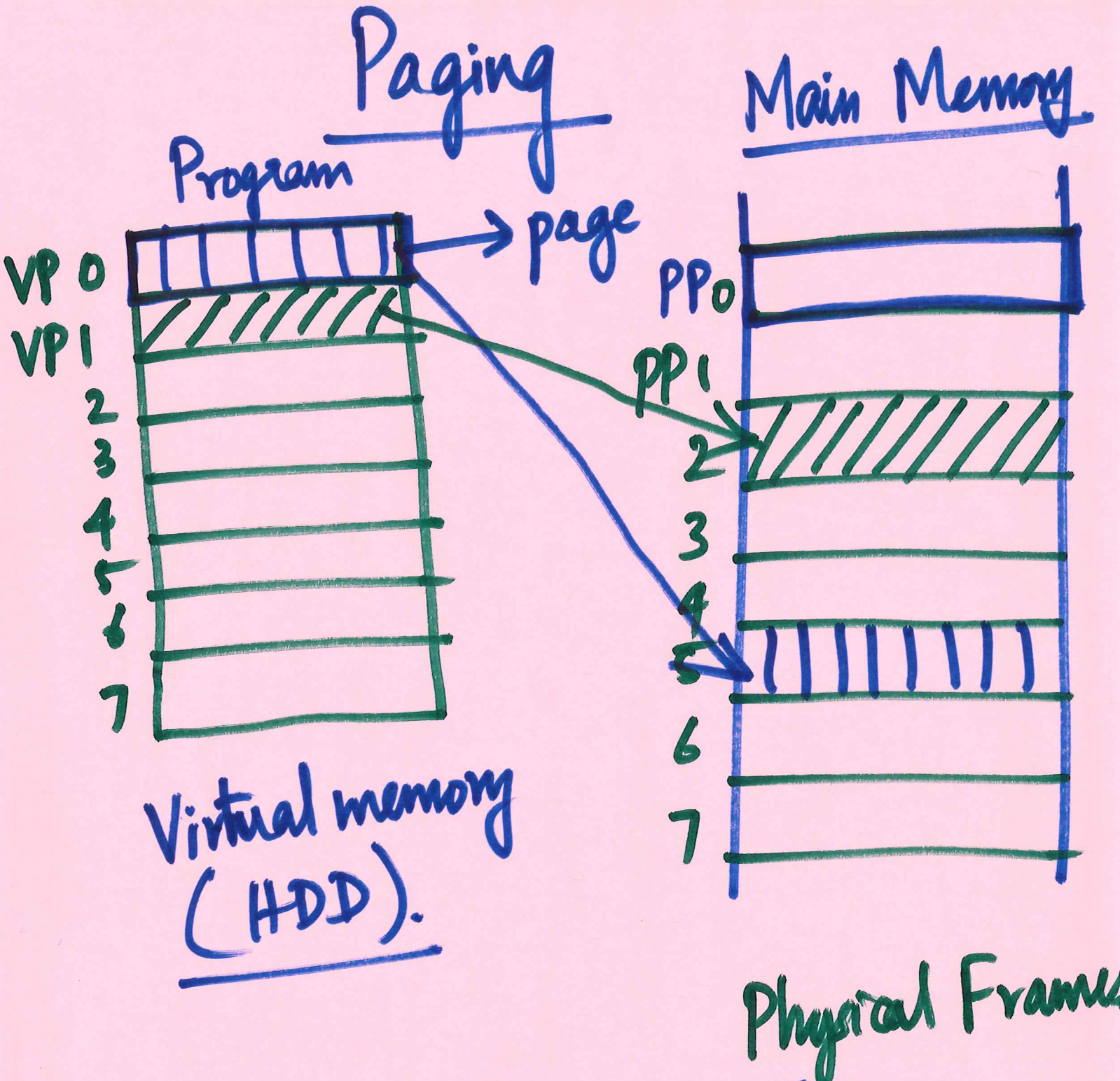


1. Entire pgm should be in RAM.

2. Segmentation





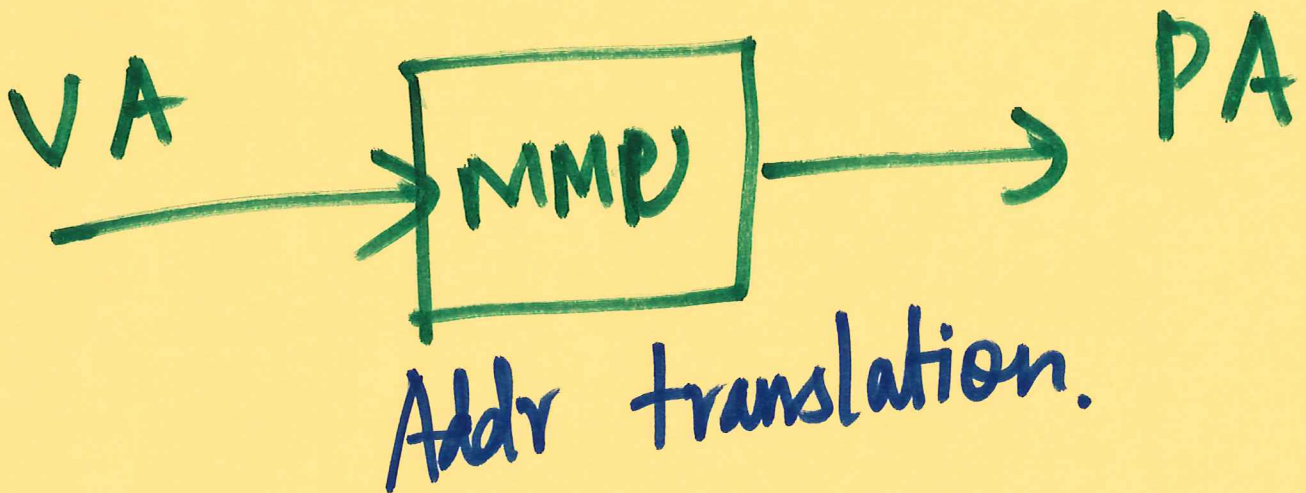


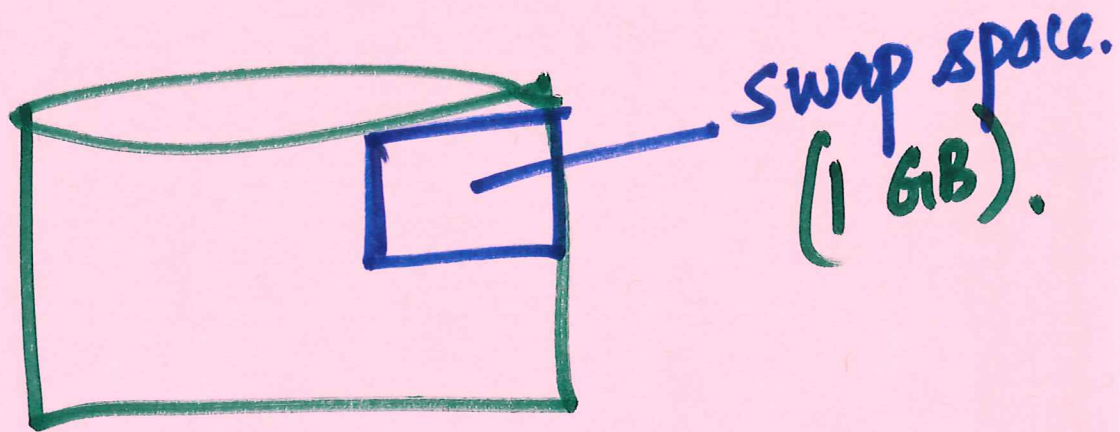
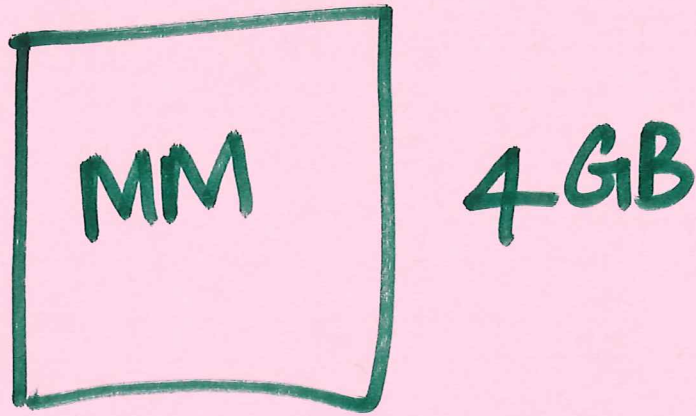
OS + Hardware
MMU

page size = 4KB

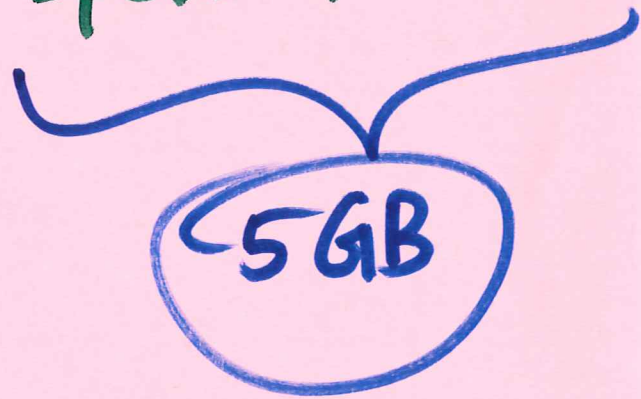
Page table - stored in main memory

VPN	PPN
0	5
1	2
2	⋮
3	⋮
4	⋮



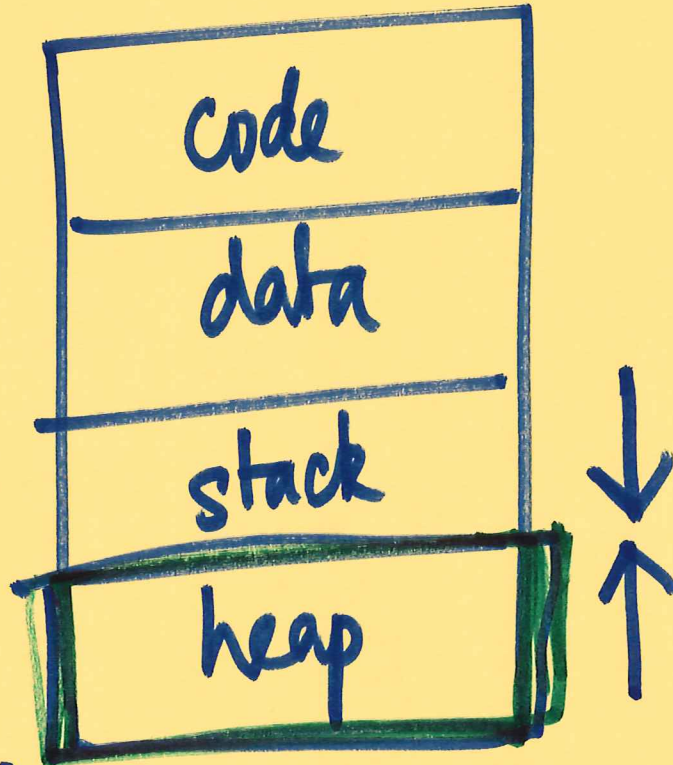


Actual MM size = 4GB + 1GB



Dynamic Memory Allocation

malloc



$n = 10;$

$\text{int } *ptr = \text{malloc}(\text{sizeof}(\text{int}) * n);$

stack

heap

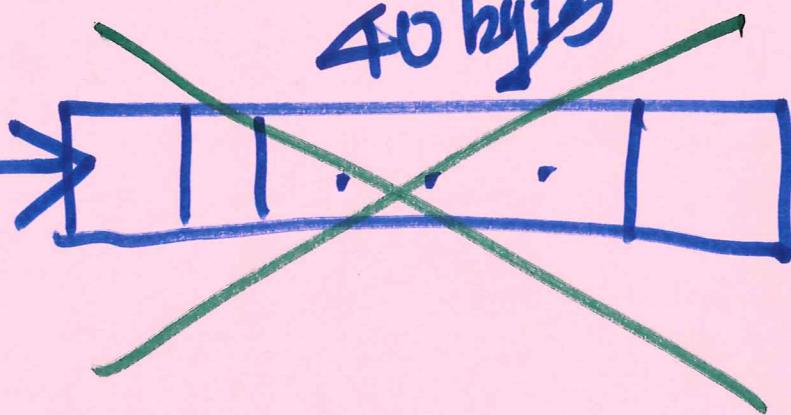
40 bytes
for 10 ints.

stack



heap

40 bytes



```
free(ptr);  
ptr = NULL;
```

```
#include <stdlib.h>
```

```
void * malloc ( size_t size );
```

↓
unsigned int

```
void free ( void * ptr );
```