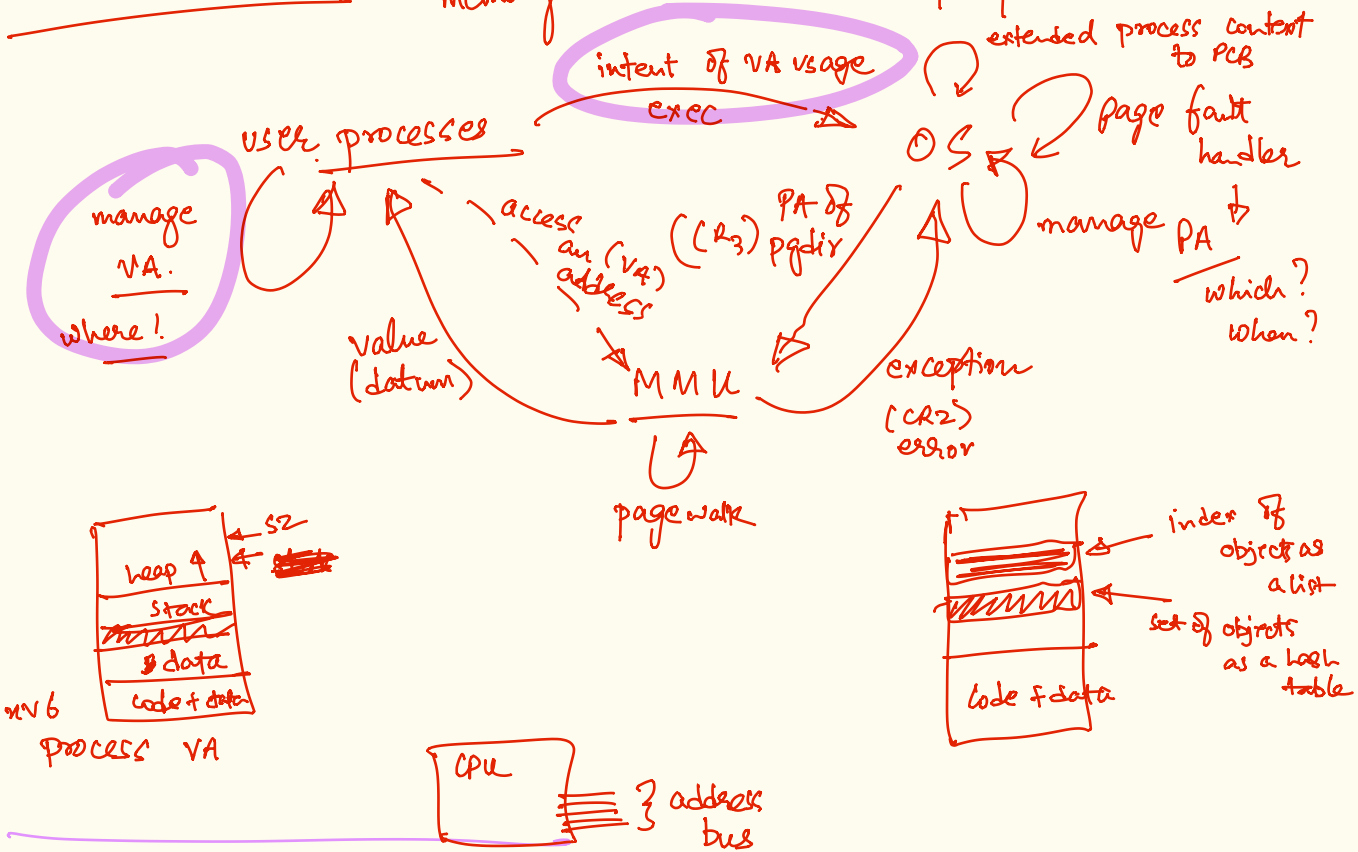


lecture 14

memory virtualization wrap-up.



(*) (i)

(*) hardware assistance for virtual memory (paging)

- MMU
- paging specific regs. CR3, CR2
- TLB (optimization)
- memory exceptions interrupt

real mode

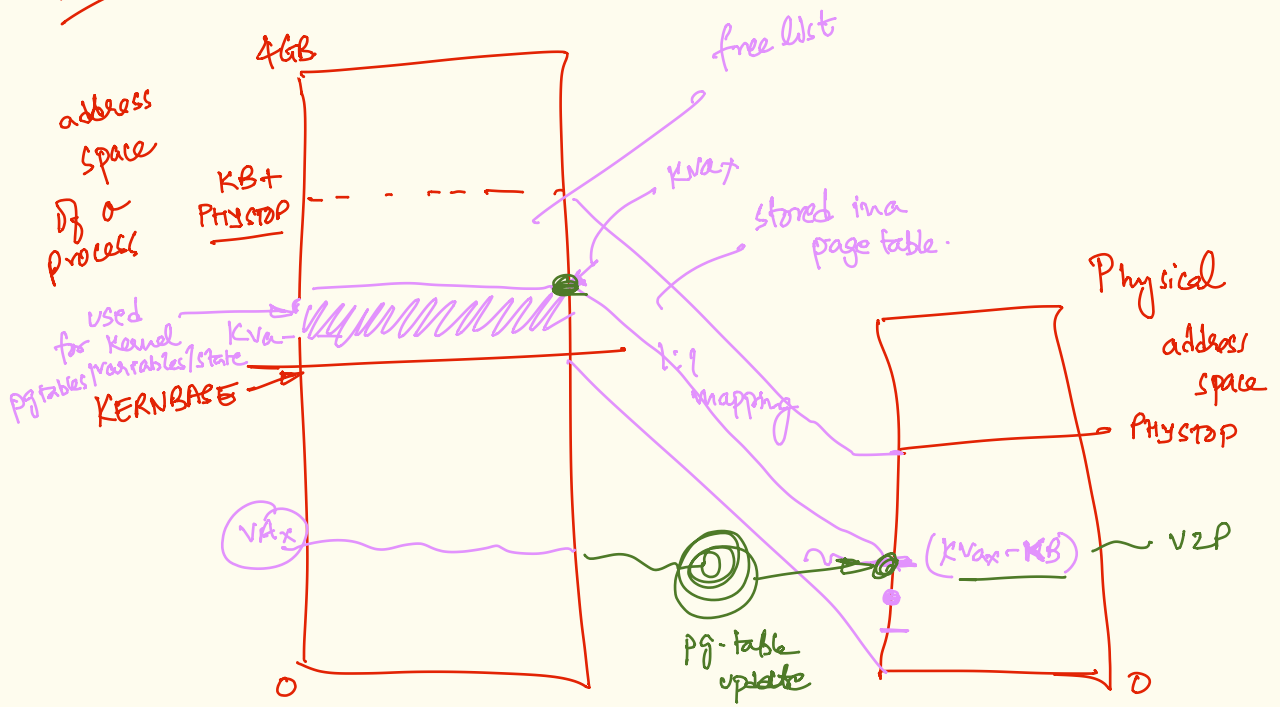
- ~ no privilege levels (no LDF)
- ~ all addressing is physical addresses.

protected mode

- ~ enables LDF instructions executed modulo privilege level check.
- ~ enables paging/segmentation $\Rightarrow LA/VA \neq PA$

x86 : CR0 → control regs. 0.
bits to set
protected mode on/off
paging on/off.

xv6 address map.

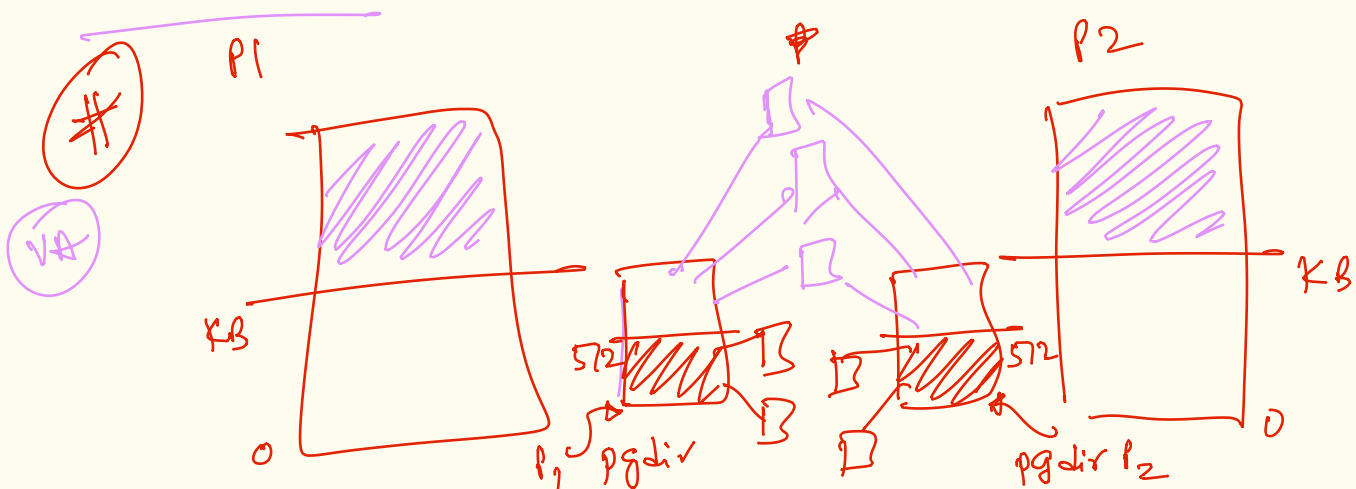


xv6.

- (i) sbrk.
 - ↳ process address space.
 - ↳ va_x with a mapped page.
 - ↳ xv6 finds a free "virtual page" in its address space \Rightarrow $kvax$ (kernel)
 - ↳ maps (va_x , V2P ($kvax$))
 - ↳ updates freelist to mark $kvax$ as active

(ii) free(va_x , #pages)

- ↳ use pg-table to find pa_x for va_x
- ↳ update freelist of kernel at P2V (pa_x) to free
- ↳ remove va_x to ba_x mapping from pgtable



VA mgmt.

```
var1 = malloc(100);
```

```
free(var1);
```

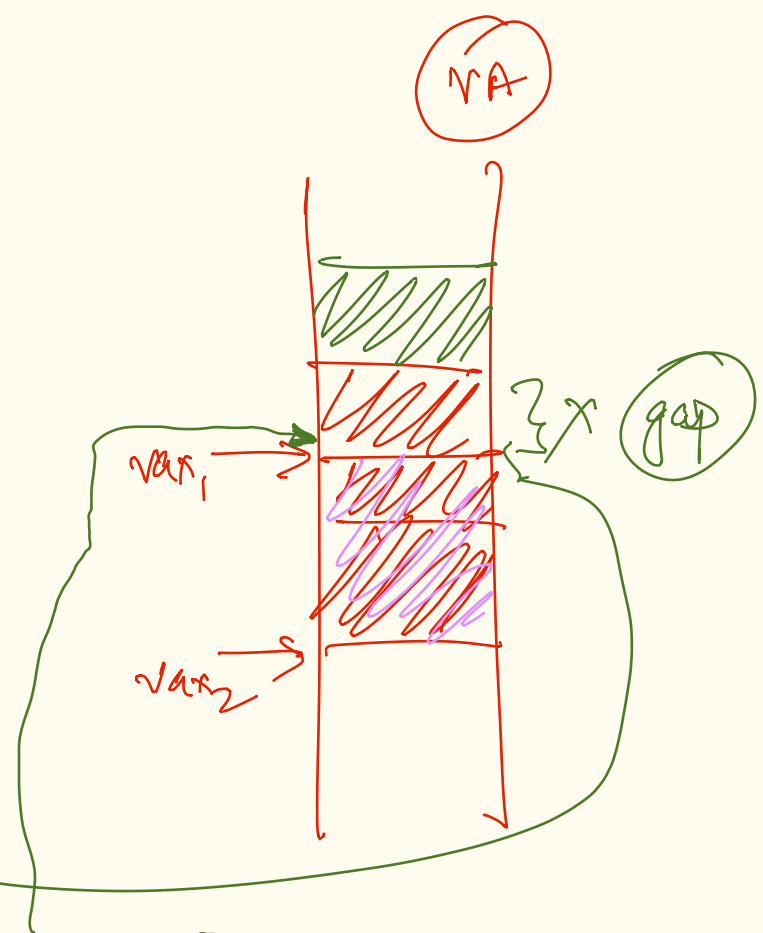
```
var2 = malloc(200)
```

```
free(var2);
```

```
free(var1);
```

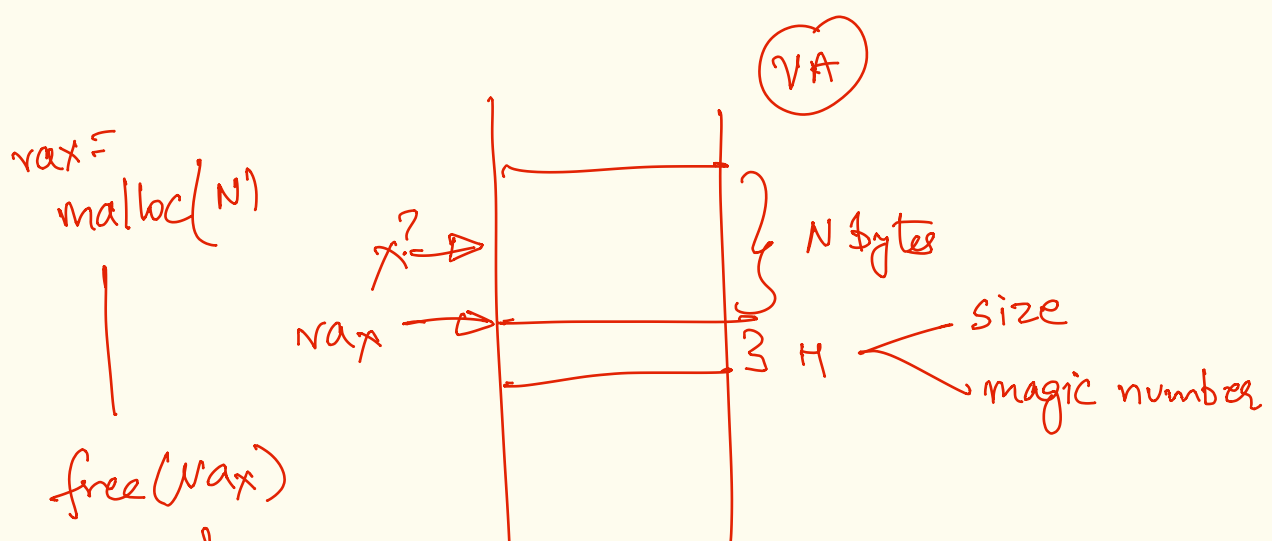
```
free(var2);
```

```
var3 = malloc(50);
```



```
va = malloc(size);
```

```
free(va); — where is the size?
```



```
size = *(var - 1);  
magic num = *(var - 2)
```

```
if (magic num != real magic)  
raise fault;
```

predefined constant