Worksheet 9

CS/ECE 354 - Spring 2016

Due: April 27th 2016 (Wednesday) in class

1. A computer has the following characteristics:
   a. 4GB Virtual Address Space
   b. 2GB Physical Address Space
   c. Page Size (P) = 4KB

   How many **bits** are required to specify the following?

   If you do not have enough information to answer the question, say "Not Enough Info".

   Virtual Page Number (VPN): 32 − VPO = 20

   Physical Page Number (PPN): 31 − PPO = 19

   Virtual Page Offset (VPO): 12

   Physical Page Offset (~~VPO~~ PPO): 12

   Maximum number of Page Table Entries (PTE) per process: 2^20

   Number of page table entries (PTE) in TLB: NEI

   Swap space (in bytes): NEI.

2. Determine the **minimum block size** for each of the following combinations of alignment requirements and block formats.

   Assumptions:
   1. Explicit free list
   2. 4-byte prev and next pointers in each free block
   3. zero-sized payloads are not allowed

   for allocated blocks. //

Need to address 4 GB.

VA → 32 bits.

Need to address 2 GB.

PA → 31 bits.

P = 4 KB.

VPO = PPO = $\log_2(4\ KB)$

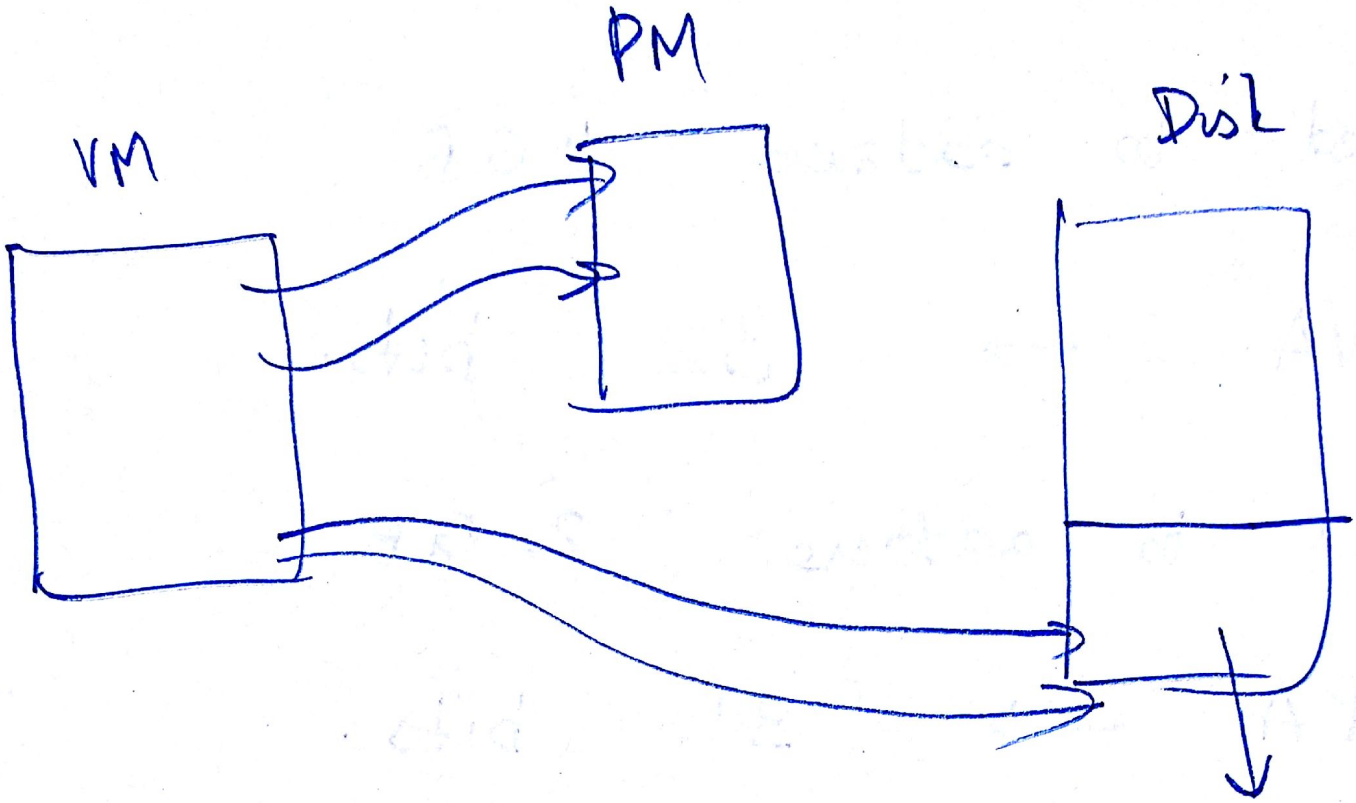
= $\log_2(4096)$

= 12 bits

1 KB = 1024 bytes

1 KB = 2^10 bytes

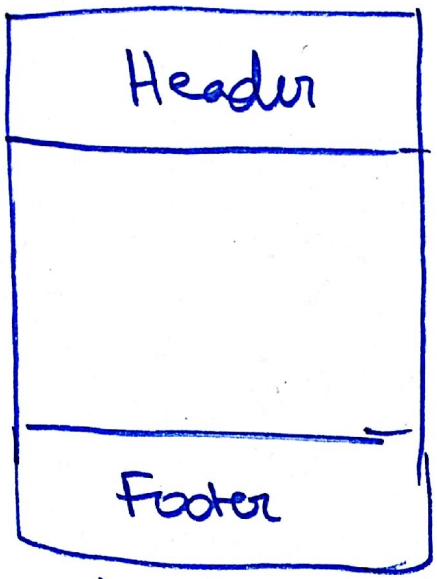
1 KB = 1000 bytes.



swap space

allocated when you set your OS up.

Allocated Block.



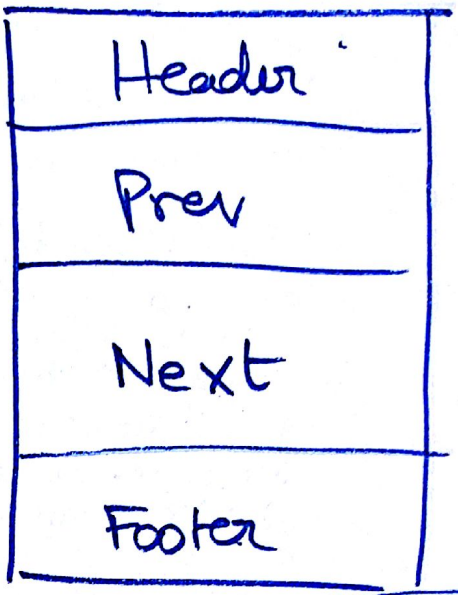
4 bytes

4 bytes

4 bytes.

12 bytes!

Free Block



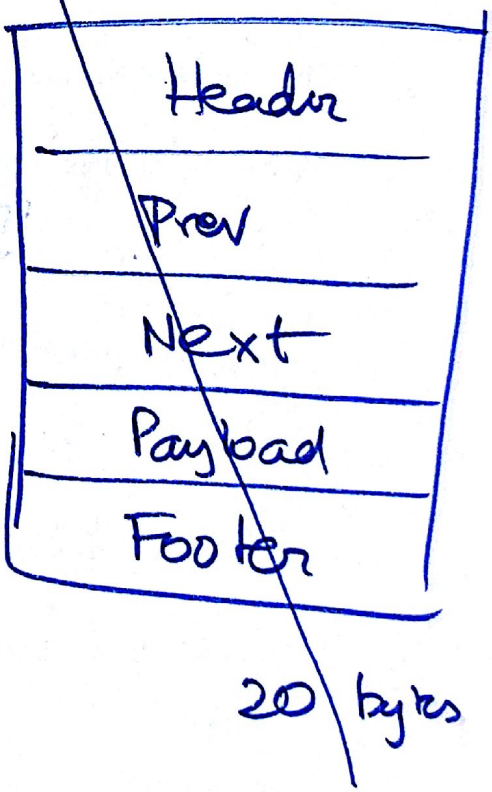
4

4

4

4.

16 byte



20 bytes

No space for payload.

Why? Because when it is freed, we get 8 bytes of space for the payload

4. headers and footers are stored in 4-byte words.
5. Block size = sizeof (header) + sizeof (payload) + sizeof (padding) + sizeof (footer) + sizeof (pointers) //.

Alignment	Allocated Block	Free Block	Minimum block size (bytes)
Single Word	Header and Footer	Header and Footer	16.
Single Word	Header but no footer	Header and Footer	—
Double Word	Header and Footer	Header and Footer	—
Double Word	Header but no footer	Header and Footer	—

3. Determine the **maximum block size** for each of the following combinations of alignment requirements and header sizes.

Assumptions:

1. Implicit free list
2. Block size = sizeof (header) + sizeof (payload) + sizeof (padding)

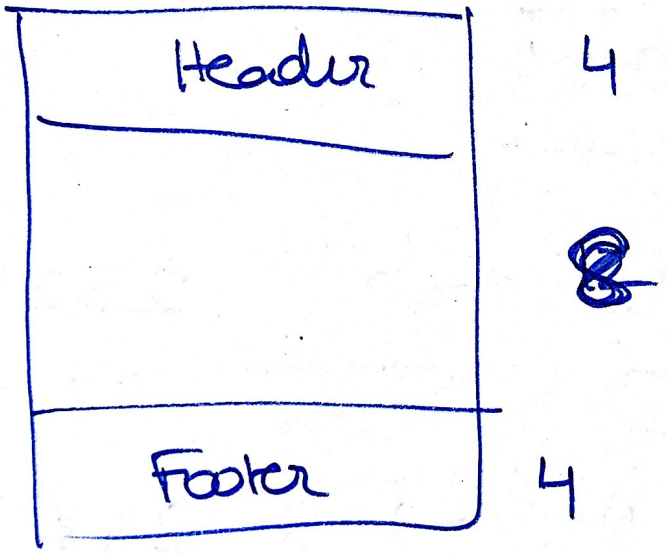
Alignment	Header Size	Maximum block size (bytes)
Single Word	1 byte	252
Single Word	2 bytes	
Double Word	1 byte	248
Double Word	2 bytes	~~248~~

1 1 1 1 1 1 1 1 → 255!

Single word → payload should start at (x 4) → side effect → each block's size is a multiple of 4.

When this allocated block is freed, will we have enough space for the prev / next pointers. //.

Allocated Block.



16.

4. headers and footers are stored in 4-byte words.
5. Block size = sizeof (header) + sizeof (payload) + sizeof (padding) + sizeof (footer) + sizeof (pointers) //.

Alignment	Allocated Block	Free Block	Minimum block size (bytes)
Single Word	Header and Footer	Header and Footer	16.
Single Word	Header but no footer	Header and Footer	—
Double Word	Header and Footer	Header and Footer	—
Double Word	Header but no footer	Header and Footer	—

3. Determine the **maximum block size** for each of the following combinations of alignment requirements and header sizes.

Assumptions:

1. Implicit free list
2. Block size = sizeof (header) + sizeof (payload) + sizeof (padding)

Alignment	Header Size	Maximum block size (bytes)
Single Word	1 byte	252
Single Word	2 bytes	
Double Word	1 byte	248
Double Word	2 bytes	~~248~~

1111 1111 → 255!

Single word → payload should start at (x 4) → side effect → each block's size is a multiple of 4.

8 — 1000

12 — 1100

16 — 10000

---

8 — 1000

12 — 1100

16 — 10000

1111 1100 → 252

---

c) 1111 1000 → 248

Last Class

1. Signals → sending / Receiving
2. Examples
3. Handling

CORRECTION

(stop signal not from terminal)

CTRL+Z ⟶✗ SIGSTOP

CTRL+Z ⟶ SIGTSTP

(stop signal is from the terminal)

Printf vs write.

MISSING A PAGE HERE from lecture today.

We talked about how signal() is not standard.

So we use POSIX

↓

sigaction().

→ portable!

→ implemented in unix, linux, OSX.

What to do when a sys call is interrupted by a signal?

How to block a signal?

Hope you took notes!

```
int sigaction (int signum,
               struct sigaction *act,
               struct sigaction *oldact
              );
```