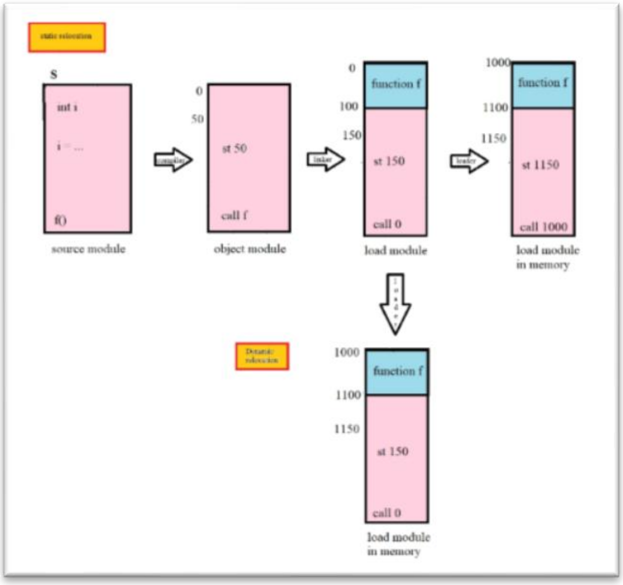
Benjamín Aage B. Birgisson

Exercise 6.1.1:

1. prior to loading
   → 50
2. after loading under static relocation
   → 150
3. after loading under dynamic relocation
   → 1150



Exercise 6.1.5:

(a)

Starting state:

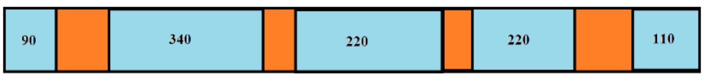
- First fit:

Block A (210) would go into hole 550

Block B (430) would then go into hole 650

Block C (100) would then go into hole 190

Block D (420) would then go into hole 420



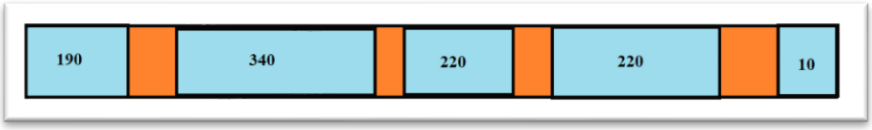
- Next fit:

Block A (210) would go into hole 550

Block B (430) would then go into hole 650

Block C (100) would then go into hole 110

Block D (420) would then go into hole 420



- Best fit:

Block A (210) would go into hole 220

Block B (430) would then go into hole 550

Block C (100) would then go into hole 110

Block D (420) would then go into hole 420

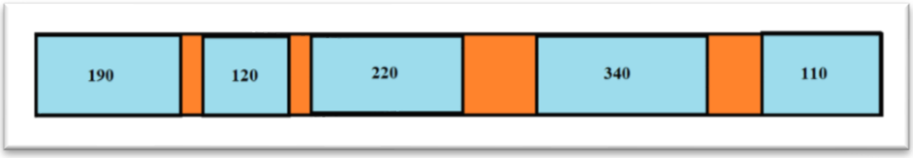
- Worst fit:

Block A (210) would go into hole 650

Block B (430) would then go into hole 550

Block C (100) would then go into hole 650 as well (after A being 440)

Block D (420) would then go into hole 420



HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

Exercise 6.2.1:

(a)

- 18 MB converts to 18,432 KB (or holes/blocks).
One third of all memory partitions are holes, by the 50% rule.
Thereby, the total number of holes are (18,432 * 0.33) 6,082.5 approximately, or 6 MB.

(b)

- 18 MB converts to 18,432 KB (or holes/blocks).
Two thirds all memory partitions are occupied blocks.
Thereby, the total number of occupied blocks are (18,432 * 0.67) 12,349.5 approximately, or 12 MB

(c)

- 18 MB converts to 18,432 KB.
One thirds of all memory partitions are holes, by the 50% rule.
As there is originally available 18,432 KB in space, then the space occupied by holes should be (18,432 * 0.3) = 6,082.5 KB approximately, or 6 MB.

Exercise 6.2.3:

(a)

f = k / (k + 2)
f = fraction of space occupied by holes
k = ratio between average hole size and average block size

<u>Occupied block = 2 * hole :</u>
f = 0.5 / (0.5+2)
f = 0.5 / 2.5
f = 0.2
f = 20%

(b)

f = k / (k + 2)
f = fraction of space occupied by holes
k = ratio between average hole size and average block size

<u>Hole = 2 *Occupied block :</u>
f = 2 / (2+2)
f = 2 / 4
f = 0.5
f = 50%

HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

Exercise 6.2.4:

(a)

- Memory size is 256 MB, which converts to 262,144 KB.

By the 50% rule, then space occupied by holes (262,144 * 0.33) is 86,507.5 MB, or the amount of holes, and space occupied by blocks (262,144 * 0.67) is 175,636.5 approximately, or the amount of blocks.
A 32-bit word equals 4 bytes (or 0.000001 MB).
Then 175,636.5 / 0.000001 = 175,636,500 bytes * 10 ns = 1,756,365,000 ns or 1.7564 seconds.

Exercise 6.3.1:

(a)

4 pages
4096 words
64 frames (physical memory)

<u>Size of logical address:</u>
Size of address space is of the form 2^n, where n is the number of bits necessary to form addresses for the entire space.
The page size is of the form 2^k, where k is the number of bits needed to address every word on a page.
The remaining n-k bits determine the number of pages, 2^{n-k}.

Number of pages: n-k = 4
Page size (bits): k = 2 bits → 00,01,10,11 (4 pages)
Address space: n = 6 → 6(n) – 2(k) = 4(n-k)

So the size of the logical address should then be 2^6.

(b)

4 pages
4096 words
64 frames (physical memory)

<u>Size of the physical address:</u>
With m bits in the physical address, 2^m total physical addresses can be formed.
The frame size must be equal to the page size and thus k bits are needed to address every word within a frame.
The remaining m-k bits determine the number of frames, 2^{m-k}.

Number of frames: m-k = 64
Frame size: k = 2
Physical space: m – 2 = 64 → m = 62 bits

HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

Exercise 6.3.2:

(a)

- The page table size (# of pages).

A: For 512 words, we need 9 bits (address range [0:111111111]). With logical address size being 16, the offset w is 7 (page table size).

B: For 1024 words, we need 10 bits (address range [0:1111111111]). With logical address size being 16, the offset w is 6 (page table size).

C: For 512 words, we need 9 bits (address range [0:111111111]). With logical address size being 32, the offset w is 23 (page table size).

D: For 1024 words, we need 10 bits (address range [0:1111111111]). With logical address size being 32, the offset w is 22 (page table size).

- The size of the logical address space (# of words).

<u>Size of logical address:</u>
Size of address space is of the form 2^n, where n is the number of bits necessary to form addresses for the entire space.

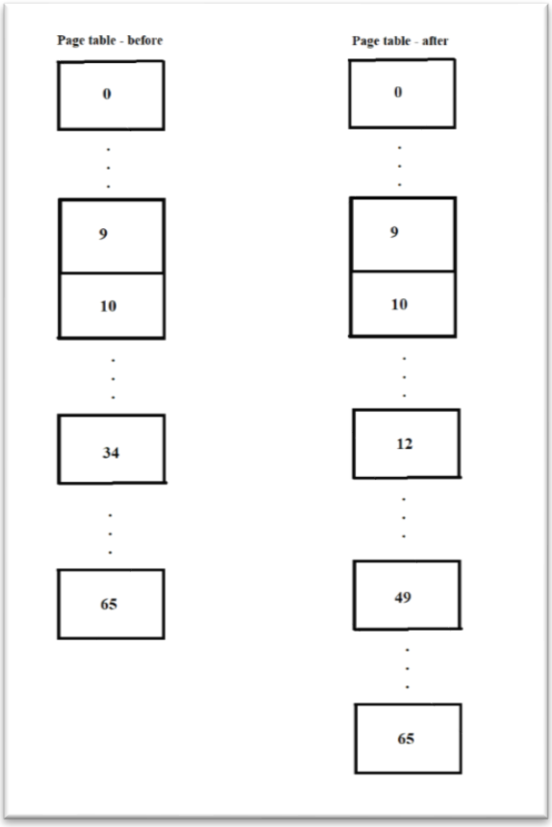
The page size is of the form 2^k, where k is the number of bits needed to address every word on a page.

The remaining n-k bits determine the number of pages, 2^{n-k}.

A: k = 9, n-k = 7 → size of logical address space = 16
B: k = 10, n-k = 6 → size of logical address space = 16
C: k = 9, n-k = 23 → size of logical address space = 31
D: k = 10, n-k = 22 → size of logical address space = 32

Exercise 6.3.3:

(a)



(b)

Logical address : (page number, offset within the page)

Higher order n-k bits specify page number p

Lower order k bits specify offset w within page

Physical address : (frame number, offset within the frame)

Higher order m-k bits specify frame number f

Lower order k bits specify offset w within frame

Given (p,w), access page table entry p

Read frame number f

Combine f with the offset w to form (f,w)

→ The physical address corresponding to the logical address

We have 65 pages (n-k), according to given physical memory. Therefore, to specify the page numbers, we need 7 bits (n-k) to represent every page in binary. So the offset, w, is therefore…. ?? and can then be allocated to the logical address in regards to the physical address and the offset:

Physical address	Logical address
4608	?
5119	?
5120	?
33300	?

Exercise 6.4.1:

(a)

Logical address 32 bits = n
K = 9 bits (512 words)
w = n-k = 23

(b)

Same as the size of w?

n = 32 bits
k = 512 words = 9 bits (11111 1111)
w = n-k → 32-9 = 23 pages

Exercise 6.4.4:

(a)

- Accessing memory m ns * (accessing TLB) / hit ratio
→ m * (m/10) / 0.5