MATH 410 502
Homework 2

Keegan Smith

September 12, 2024

Problem 1

Initially our memory space looks like:
16 byte free list: [0, ..., 15]
8 byte free list: []
4 byte free list: []
2 byte free list: []
1 byte free list: []
after 1 byte is requested:
16 byte free list: []
8 byte free list: [8,..., 15]
4 byte free list: [4,..., 7]
2 byte free list: [2, 3]
1 byte free list: [1]
after 2 byte requested (so now 3 bytes in total requested):
16 byte free list: []
8 byte free list: [8,..,15]
4 byte free list: [4,..., 7]
2 byte free list: []
1 byte free list: [1]
after 4 byte requested (so now 7 bytes in total requested):
16 byte free list: []
8 byte free list: [8,..,15]
4 byte free list: []
2 byte free list: []
1 byte free list: [1]
after 2 byte requested (so now 9 bytes in total requested):
16 byte free list: []
8 byte free list: []
4 byte free list: [12,...,15]
2 byte free list: [10,11]
1 byte free list: [1]

Problem 2

a) There are $\frac{2^{36}}{2^{13}} = 2^{23}$ pages in the virtual address space.

b) If each page table entry is 4 bytes $= 2^5 = 32$ bits, then there must be 2^{32} physical pages. Thus the total addressable memory should be that times the size of a page: $2^{32} \cdot 2^{13} = 2^{45}$ bytes.

c) I would use a 1 level page table because an 8GB process would be using most of the memory anyways. A 2 or 3 level page table would only serve to obfuscate things as you would still have to allocate memory for pretty much all of the 2nd and 3rd level pages.

d) For a single page table, the number of page table entries would be $\frac{2^{3}3}{2^{1}3} = 2^{20}$.

Problem 3

a) The page size would be 2^8 bytes

b) A process that has 2^{18} bytes would be using $\frac{2^{18}}{2^8} = 2^{10}$ pages. This means that the number of allocated page table entries in the third level table must be 2^{10}.
Each third level page table has 2^6 entries, so there must have been $\frac{2^{10}}{2^6} = 2^4$ 2nd level page entries allocated (because 2^4 third level pages were allocated).
The size of a level 2 page table is 2^8 entries, since only 2^4 entries were needed, we only allocated one level 2 page table.
This means we only needed one page entry in the level 1 page table, and thus we only allocated one level one page table. So in total, we allocated 3 level 3 tables, one level 2 table, and one level one table.
The total number of pages is $\frac{2^{32}}{2^8} = 2^{24}$, so the size of a page entry is 24 bits or 3 bytes.
So in total, the page table size was $2^6 \cdot 2^4 \cdot 3 + 2^8 \cdot 3 + 2^{10} \cdot 3$ bytes, and wasted $(2^8 - 3 + 2^{10} - 1) \cdot 3$ bytes due to internal fragmentation

c) For the code segment, we are allocating a total of $2^{14} \cdot 3$ bytes which is $\frac{2^{14} \cdot 3}{2^8} = 2^6 \cdot 3$ pages. Thus we need $2^6 \cdot 3$ entries in our L3 page table, L3 has 2^6 entries per table, so we allocated 3 tables. Since we allocated 3 tables in L3, we must've used 3 entries in L2 page table, the number of entries in an L2 table is 2^8 so we only needed to allocate 1 L2 page table. So of course we only allocated 1 L1 page table. So for the code segment's

page tables we consumed: $(2^6 \cdot 3 + 2^8 + 2^10) \cdot 3$ bytes. Memory which was not used (in the L1 and L2 page tables) is $(2^8 - 3 + 2^{10} - 1) \cdot 3$ (we only used 3 entries in the L2 table and only one entry in L1).
For the data segment: $600K = 75 \cdot 2^{15}$ bytes $= \frac{75 \cdot 2^{15}}{2^8}$ pages $= 75 \cdot 2^7$ pages. So we will need $\frac{75 \cdot 2^7}{2^6} = 75 \cdot 2$ L3 page tables. So we will need $\lceil \frac{75 \cdot 2}{2^8} \rceil = 1$ L2 page (which is wasting $(2^8 - 150) \cdot 3 = 318$ bytes). We are only using 1 L1 entry so 1 L1 page table, and so we are wasting $(2^{10} - 1) \cdot 3$ bytes.
In total for the data segment, we allocated: $(75 \cdot 2 \cdot 2^6 + 2^8 + 2^{10}) \cdot 3$ bytes. We wasted $318 + (2^{10} - 1) \cdot 3$ bytes.
For the stack segment, I already did this in b). We allocated a total of $2^6 \cdot 2^4 \cdot 3 + 2^8 \cdot 3 + 2^{10} \cdot 3$ and wasted $(2^8 - 3 + 2^{10} - 1) \cdot 3$ bytes.
So in total we allocated $(2^6 \cdot 3 + 2^8 + 2^{10}) \cdot 3 + (75 \cdot 2 \cdot 2^6 + 2^8 + 2^{10}) \cdot 3 + 2^6 \cdot 2^4 \cdot 3 + 2^8 \cdot 3 + 2^{10} \cdot 3$ for page tables and wasted $(2^8 - 3 + 2^{10} - 1) \cdot 3 + 318 + (2^{10} - 1) \cdot 3 + (2^8 - 3 + 2^{10} - 1) \cdot 3$ bytes due to internal fragmentation.

Problem 4

a) The size of the page should just be the max page offset, 2^{12} bytes, since the protection bits should be accounted for in the page table entries.

b) For a process that uses 64K memory we have: $64K = 2^6 \cdot 2^{10} = 2^{16}$, dividing this by the size of a page should give us the total number of pages needed by the program:

$$\frac{2^{16}}{2^{12}} = 2^4 \text{ pages}$$

So we will be needing 2^4 entries in our L3 page table. There are 2^8 entries per L3 page table, so we should be fine with just one $L3$ page table. This will waste $2^8 - 2^4$ entries.

We only needed one L3 page table, so we will only need one L2 entry, and thus only one L2 page table. The number of entries in an L2 page table is 2^4, so we will be wasting $2^4 - 1$ entries.

We only needed one L2 page table, so we will only need one L1 entry, and thus only one L1 page table. The number of entries in an L1 page table is 2^8, so we will be wasting $2^8 - 1$ entries.

In total we used 1 L3 page table, 1 L2 page table, and 1 L1 page table. So in total we allocated $2^8 + 2^4 + 2^8 = 2^9 + 2^4$ page table entries. Each page table entry is $44 + 4$ bits (physical address + protection bits) which is 6 bytes. So in total, the page tables used $(2^9 + 2^4) \cdot 6 = 3168$ bytes.

In total, we wasted: $2^8 - 2^4 + 2^4 - 1 + 2^8 - 1 = 2^9 - 2$ page table entries

which is $(2^9 - 2) \cdot 6 = 3060$ bytes.

c) For the code segment, we are allocating a total of $2^{14} \cdot 3$ bytes which is $\frac{2^{14} \cdot 3}{2^1 2} = 2^2 \cdot 3$ pages. Thus we need $2^2 \cdot 3$ entries in our L3 page table, L3 has 2^8 entries per table, so we allocated 1 table. This will waste $2^8 - 2^2 \cdot 3$ entries.

We only allocated 1 L3 table, so we only need one L2 entry, and thus only one L2 table. This will waste $2^4 - 1$ entries.

We only allocated 1 L2 table, so we only need one L1 entry, and thus only one L1 table. This will waste $2^8 - 1$ entries.

So for the code segment page tables, we allocated

$$2^8 + 2^4 + 2^8 = 2^9 + 2^4 \text{ entries}$$

entries, for a total of $(2^9 + 2^4) \cdot 6 = 3168$ bytes. We wasted:

$$2^8 - 2^2 \cdot 3 + 2^4 - 1 + 2^8 - 1 = 514 \text{ entries}$$

for a total of $514 \cdot 6 = 3084$ bytes wasted.
For the data segment: $600K = 75 \cdot 2^{15}$ bytes $= \frac{75 \cdot 2^{15}}{2^1 2}$ pages $= 75 \cdot 2^3$ pages. So we will need

$$\lceil \frac{75 \cdot 2^3}{2^8} \rceil = 3$$

L3 page tables, which wastes $3 \cdot 2^8 - 75 \cdot 2^3 = 168$ entries.

So we will need $\lceil \frac{3}{2^4} \rceil = 1$ L2 page (which is wasting $2^4 - 3$ entries).

We are only using 1 L1 entry so 1 L1 page table, and so we are wasting $2^8 - 1$ entries.

In total for the data segment page tables we are allocating:

$$3 \cdot 2^8 + 2^4 + 2^8 = 1040 \text{ entries}$$

So we are allocating $1040 \cdot 6 = 6240$ bytes. We are wasting:

$$168 + 2^4 - 3 + 2^8 - 1 = 436 \text{ entries}$$

which is $436 \cdot 6 = 2616$ bytes.

We've already done the calculations for 64K in b.). Recall we allocated 3168 bytes for page tables, and wasted 3060 bytes.

So in total, for page tables, this program will allocate:

$$3168 + 6240 + 3168 = 12576 \text{ bytes}$$

And will waste:

$$3084 + 2616 + 3060 = 8760 \text{ bytes}$$