# CS 326-02 Sketch, Local Dev, Adv Heap

## Bootstrapping

The problem: sh needs a working malloc/free > 4096

we have

```
void free(p)
void * malloc_name(nbytes name)
void * malloc(nbytes)
```

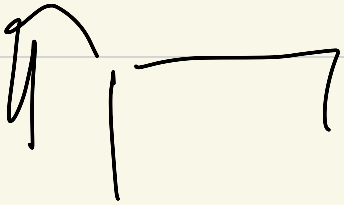
↓

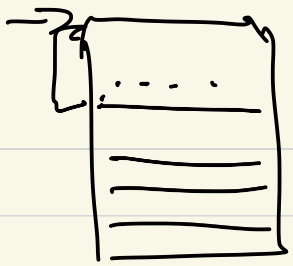
```
add  malloc_name_given()
have  malloc()
        call  malloc_name_given()

add  myfree()  (user.h)
change memtest to call
      myfree()
```

```
               int
p = malloc(nbytes)

        ↘

      malloc_name(nbytes, name)

free(void *p)
```

---

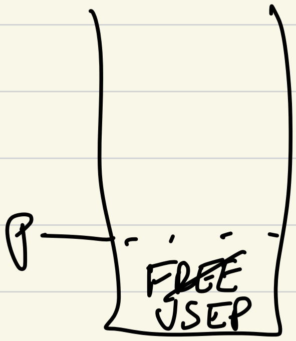
```
malloc_name(nbytes , name)
{
    if (!heap_initialized) {
      ....
    }

    if (nbytes <= BLOCK_MIN_SIZE)
        nbytes = BLOCK_MIN_SIZE;

    fp = find_free_block(nbytes)

    if (fp == 0) {    // not found

        fp = request_more_heap()
    }

    p = split_free_block(fp, nbytes)

    return p;

}
```





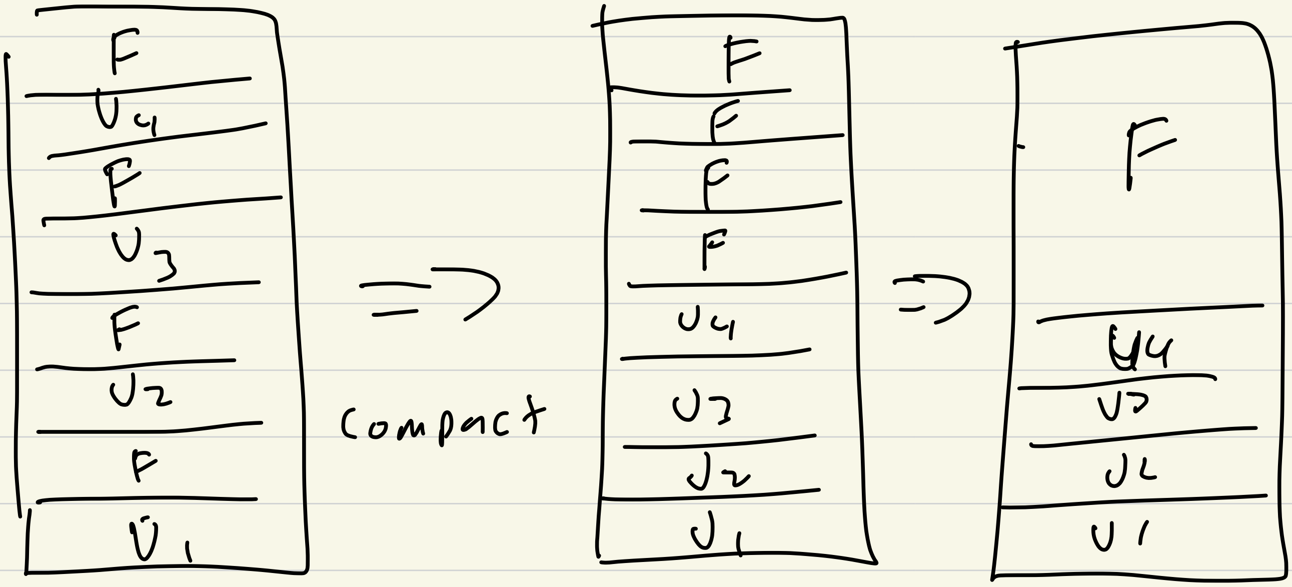
```
request_more_heap(nbytes)
{
    amt = nbytes + sizeof(struct block_hdr)

    if (last block in bloch_list is FREE)
        amt = amt - (sizeof(struct bloch_hdr)
                  + last_free->size)

  // Determine request size
  factor = amt / BLOCK_MIN_INCR
              + Remainder

new_heap = sbrk(factor * BLOCK_MN_INR)

    make new free block, fp starting at
      new_heap

    if (last block in bloch_list is FREE) {
      fp merge(last_free, fp)
    }

    return fp;

}
```

```
void *
split_free_block (fp, nbytes).
{
    if (we can split fp) {
        fp becomes used
        update size
        make a new free block
        add new free block to block_list
    } else {
        turn fp into USER block
          keep size
    }
    in fp determine p for user

    return p;

}
```



```
free( void *p )
{
    hp = p - (sizeof(struct block_hdr)
    set hp->used to 0
    set hp->name to empty string

    hp = merge( hp , list_next(hp) )
         merge ( list_prev(hp) , hp )
}
```

heap allocation



Python / Java

Garbage Collection

```
f = new Foo()
↑
```

object reference