Feb 17, 2021

HW3 due Tues 2/23 by 11pm

Today
- recap questions
- review prev. exercises
- stack allocation
- heap allocation
- ex4-2 *
  Note: has been updated, you may need to pull from public repo again

slido.com
jhuip01
zero

---

Recap Qs

1) Stack: local variables

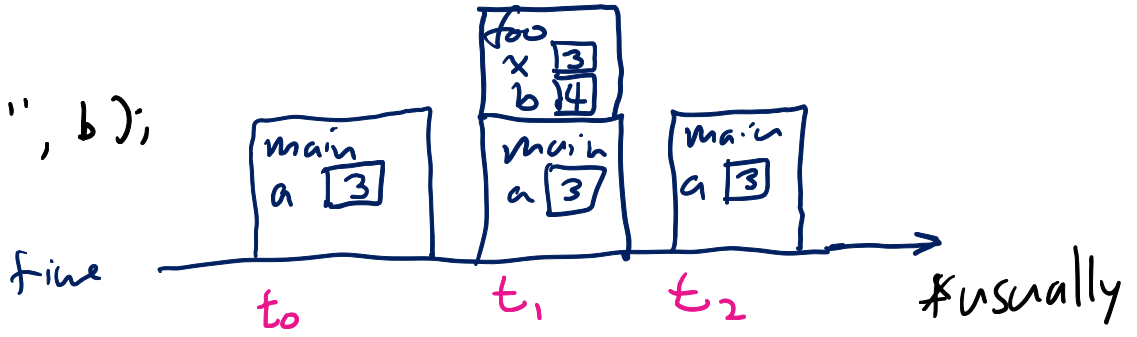
call stack
→ stack frames
call: push
returns: pop

variable: named* storage loc.

```
main() {
    int a;
    a = 3;
    foo(a);
}

foo(int x) {
    int b;
    b = x + 1;
    printf("%d", b);
}
```

t0 t1 t2



Heap memory: allocate memory (var, array)
(malloc/free) with arbitrary life time
access indirectly using pointers

---

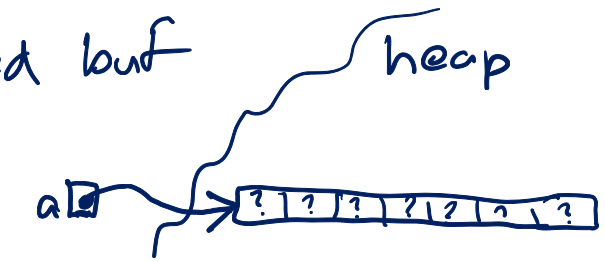
Stack memory: automatically alloced/dealloced

Heap memory: must be explicitly dealloced
(Java: new)

2) malloc – allocate buffer (range of memory addresses) dynamically (returns pointer)
free – dealloc buffer

3) memory leak
alloc w/ dealloc
} ⟹ valgrind

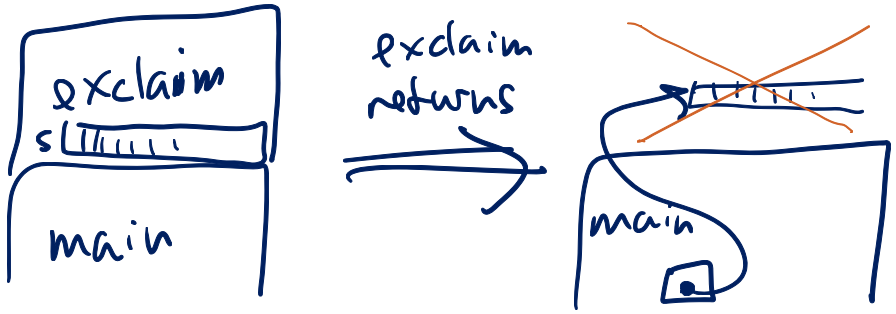
4) malloc – allocate unitialized memory
calloc – alloc 0-filled buf
realloc – "resizes" buffer




```
int *a;
int nitems;
    :
a = (int*) malloc (sizeof(int)
                   * nitems);
```

5) valgrind → invalid accesses (reads/writes)
array index OOB
→ memory leaks

6)



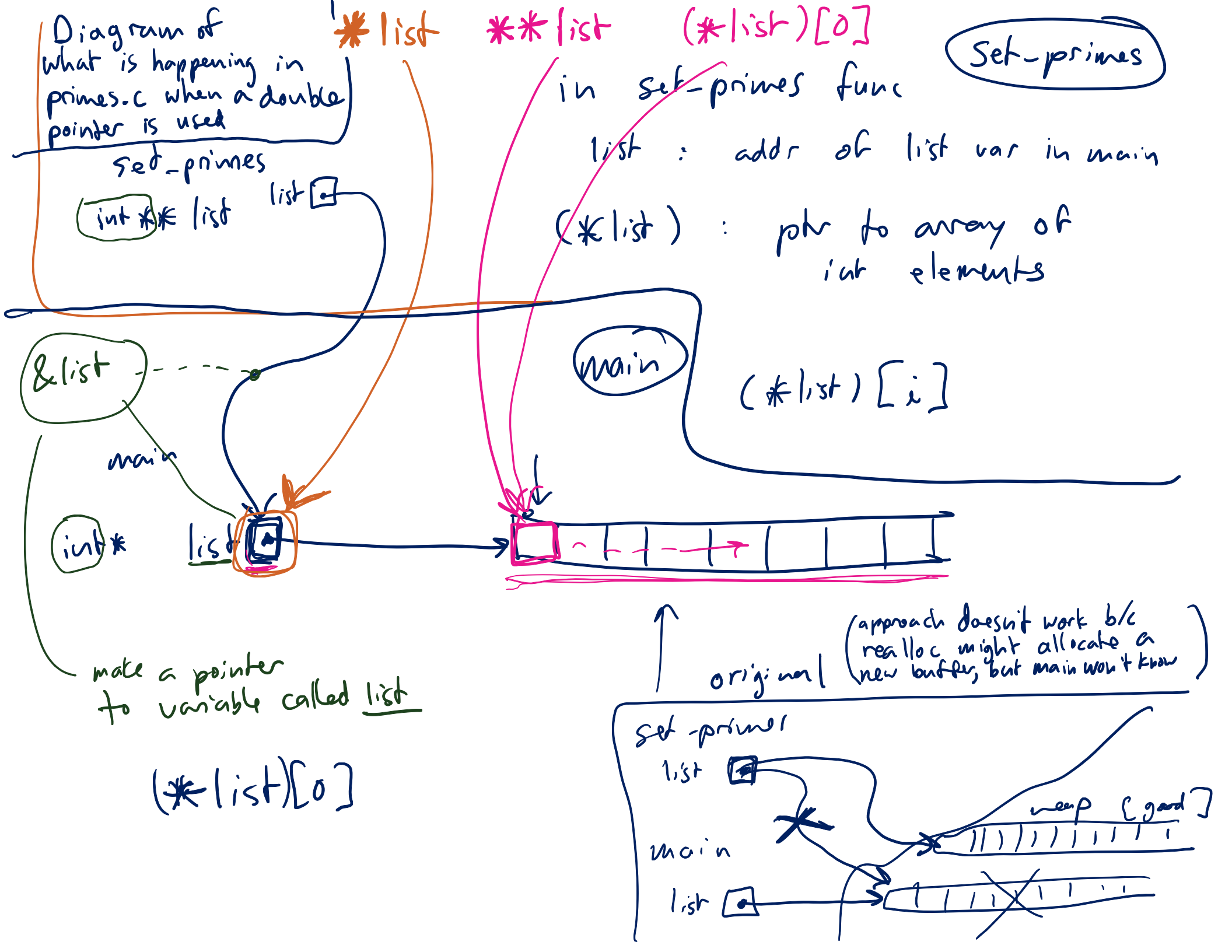


Diagram of what is happening in primes.c when a double pointer is used
*list
**list
(*list)[0]
set_primes
in set_primes func
list : addr of list var in main
(*list) : ptr to array of int elements
set_primes
int** list
list
&list
main
(*list)[i]
main
int*
list
make a pointer to variable called list
(*list)[0]
original
(approach doesn't work b/c realloc might allocate a new buffer, but main won't know
set_primes
list
temp [guard]
main
list