November 1, 2021

Lab #4

45 - Deck

[1] .... [1]

45 decks

Move 1 : [0] ... [0] [45]

46 array

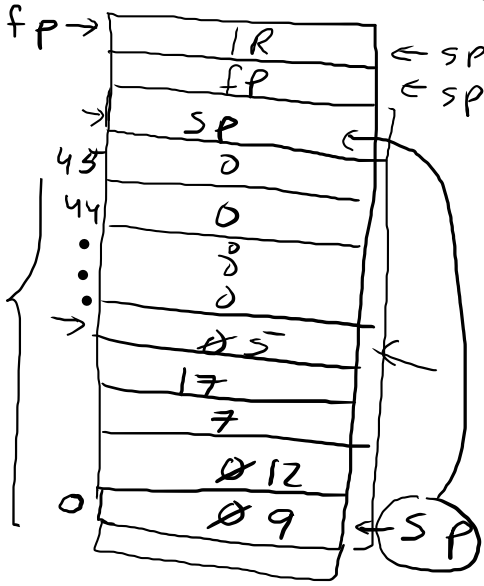
45

46 decks

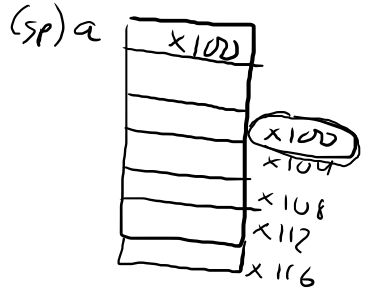
Decks

main.s

Stack



int a[5];



ldr r0, [fp, #-8]

address of 1st elt array

sub sp, sp, #188

47
*4
188

ran_allocate

1. gen 1 → 45 ⇒ 9

↦ 36 → 12 1 → 24 → 7

[9] [12] [7] [17]

end game

r4 = ndeck = 4

2. play-game new_deck = 4 r4 = 5

[8] [11] [6] [16] [4]

check_endgame

[sp, #(ndecks-1*4)]

9 decks

[] ... []

2 1 9 7

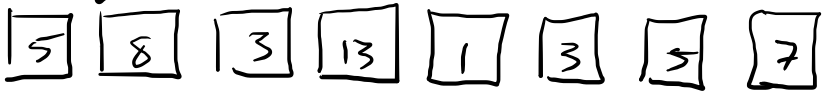
1 2 3 ... 8 9

ndecks = 5

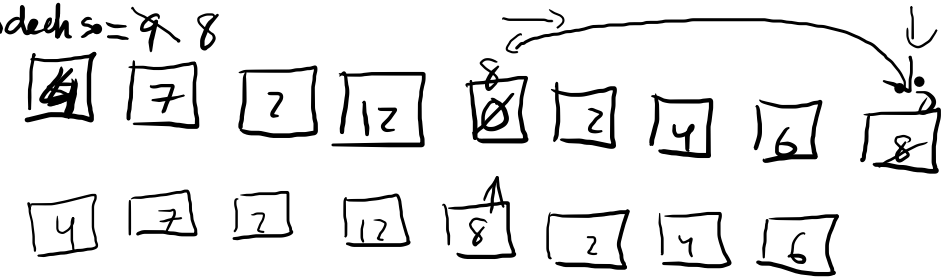
ndeck = 6

ndeck = 7

ndecks = 8



ndecks = ~~9~~ 8




```
while (! endgame(a, ndecks))
{
    (1) make_move
    (2) remove_zero
}
```

remove_zero

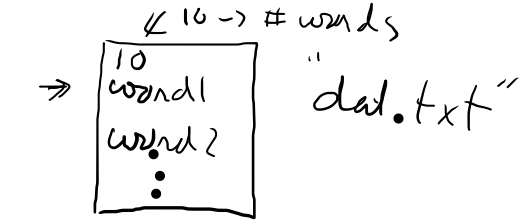
Continue from last week

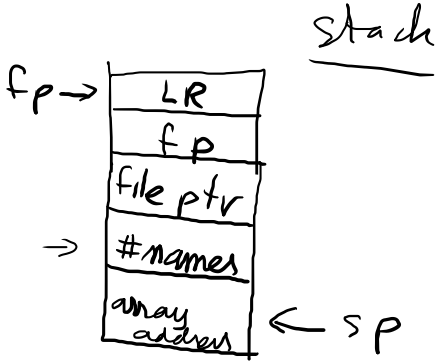
- main.s
- readmeta.s →

→ fp − 8 → file ptr
fp − 12 → #names
fp − 16 → address of 1st elt of my array on heap space

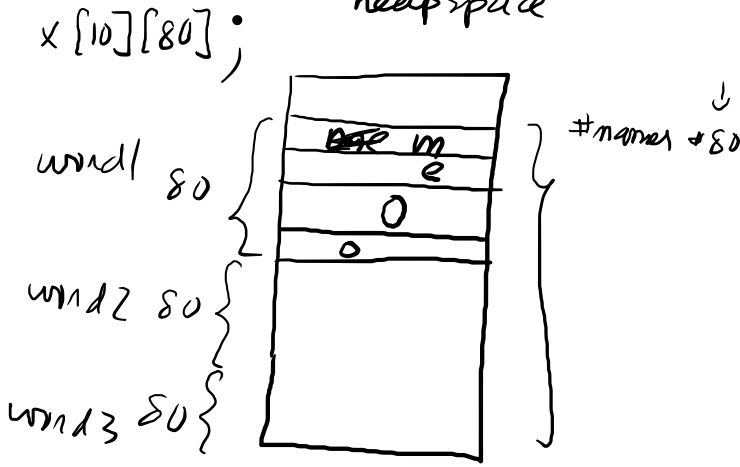
bl malloc @ allocate memory for the array
push {r0}
@ read_data(file ptr (r0), #words (r1), array address (r2))

ldr r0, [fp, #-8] @ r0 ← file ptr
ldr r1, [fp, #-12] @ r1 ← #names
ldr r2, [fp, #-16] @ r2 ← array address
bl read_data





char x[10][80];



```
@ print out names
@ print_names (array address, #names)

ldr r0, [fp, #-16]

ldr r1, [fp, #-12]

bl print_names
ldr r0, [fp, #-8]
bl fclose  @ fclose (file ptr)
sub sp, fp, #4
pop {fp, pc}
```

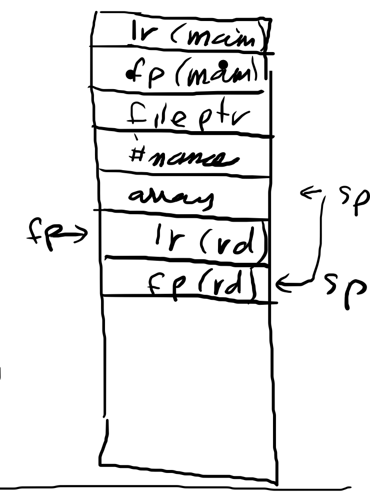
read_data.s

```
.cpu    cortex-a53
.fpu    neon-fp-armv8

.data
str_inp:    .asciz    "%s"

.text
.align 2
.global read_data
.type read_data, %function

read_data:
    push {fp, lr}
    add fp, sp, #4
```

```
@ r0 <- file ptr
@ r1 <- #names
@ r2 <- array address
@ fscanf(fileptr, "%s", sp)
@ need to save r0, r1, r2
mov r4, r0
mov r5, r1
mov r6, r2
mov r10, #0   @ i
read_loop:
    cmp r10, r5   @ i < #names
    bge end_read_loop
    mov r0, r4
    ldr r1, =str_inp
    mov r2, #80
    mul r2, r2, r10
    add r2, r2, r6
```

names[][80],

stack

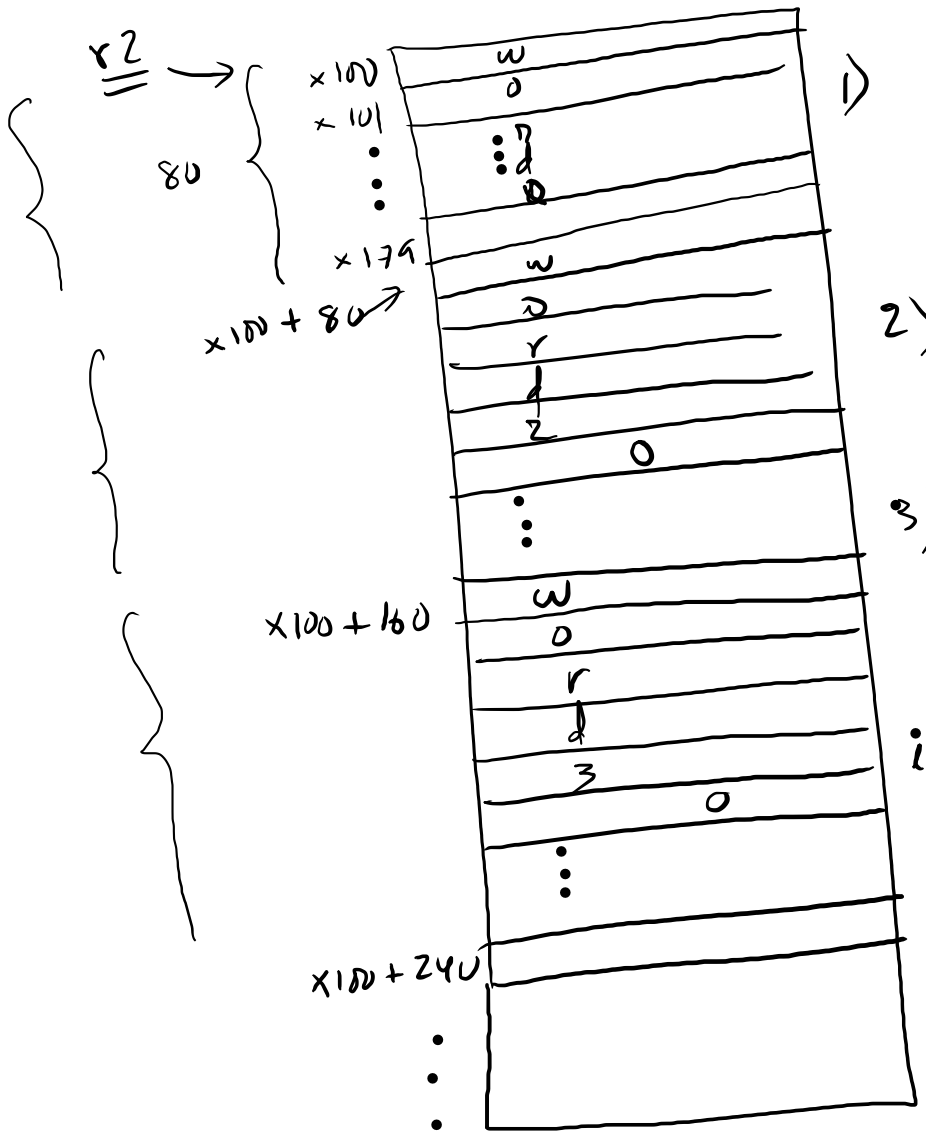



```
for (i=0; i < #names; i++)
{
                 r0          r1
    fscanf(fileptr, "%s",
                                 names + i*80)
}
```

```
    bl fscanf
    add r10, r10, #1    @ i++
    b read_loop
end_read_loop:

    sub sp, fp, #4
    pop {fp, pc}

@ done with read_data.s
```

print_names.s

```
.cpu cortex-a53
.fpu neon-fp-armv8

.data
str_out2: .asciz "%s"

.text
.align 2
.global print_names
.type print_names, %function
print_names:
    push {fp, lr}
    add fp, sp, #4
    @ r0 <- array
    @ r1 <- #names
    mov r4, r0  @ saving array
    mov r5, r1  @ and #names

    mov r10, #0
print_loop:
    cmp r10, r5
    bge end_print_loop
    @ printf("%s", name + i*80)   (r4), r10
    mov r1, #80
    mul r1, r1, r10  @ r1 = r10*80 + r4
    add r1, r1, r4
    bl printf
    add r10, r10, #1
    b print_loop

end_print_loop:
    sub sp, fp, #4
    pop {fp, pc}
```



1) fscanf(fp, "%s", x100)

2) fscanf(fp, "%s", x100 + 80)

3) fscanf(fp, "%s", x100 + 160)

⋮

i) fscanf(fp, "%s", x100 + i*80)

(r2 + i*80)

char anames[10][80];