# CSE 531: Distributed and Multiprocessor Operating Systems

Class: 11 Date: 9/30

Fall 2020

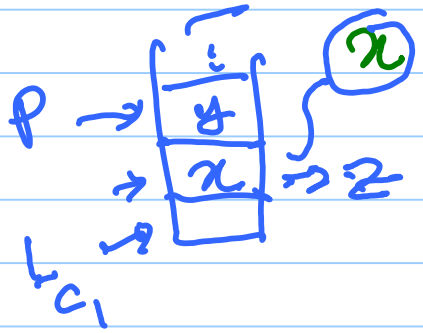
Note Title

## Coroutines

1 thread

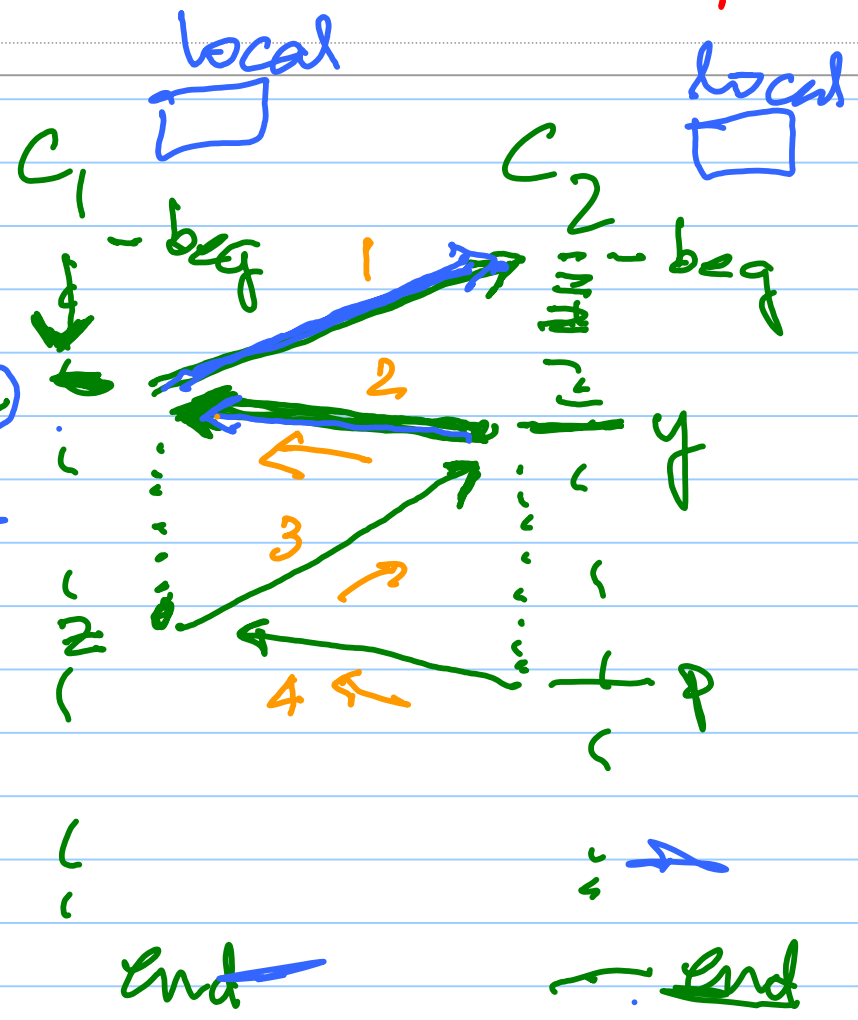
multiple coroutines

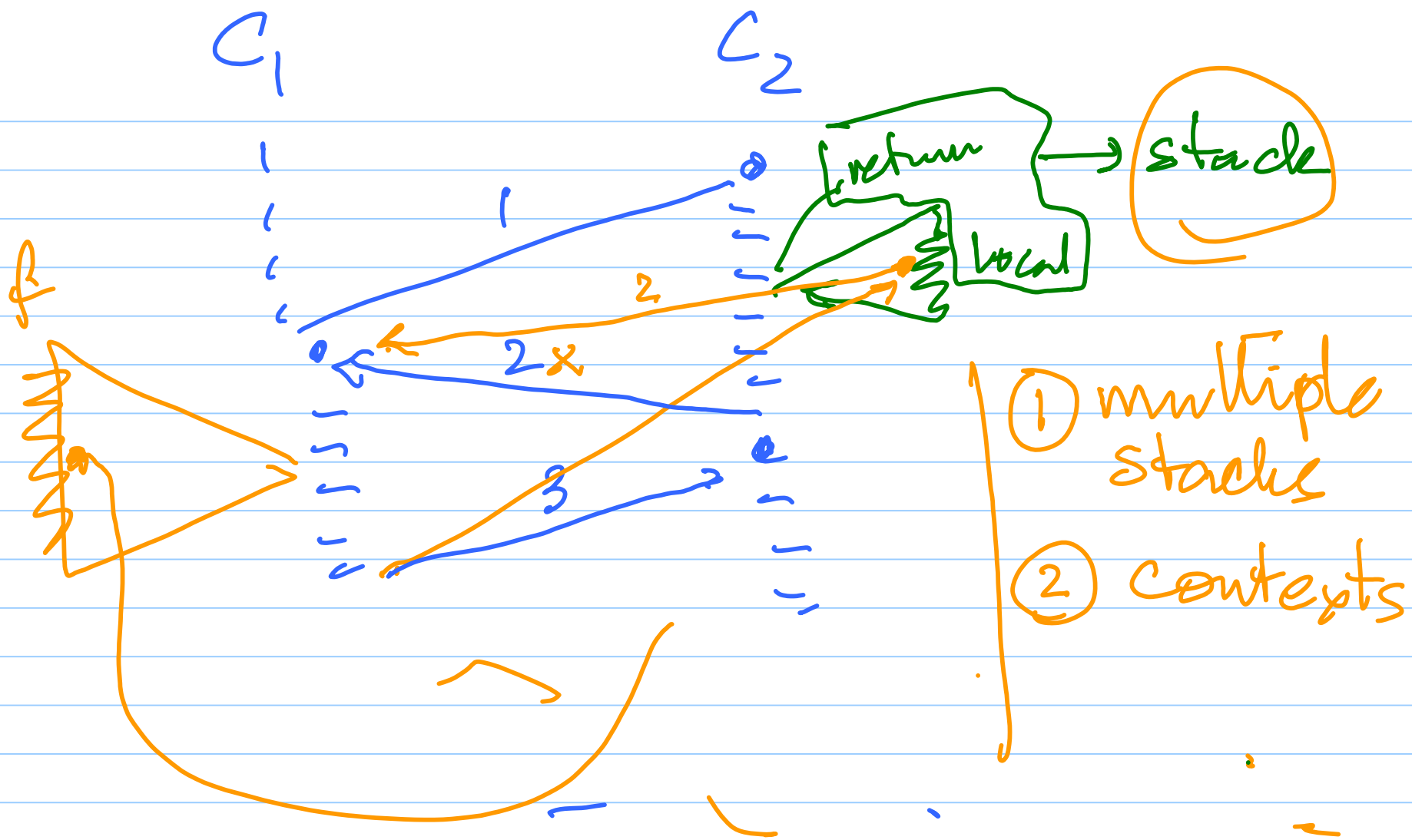
↳ (2)




Stacks → return addr

→ local variables




$C_1$
$C_2$
f
1
2
2x
3
return
local
stack
① multiple stacks
② contexts

multiple stacks → one per coroutine

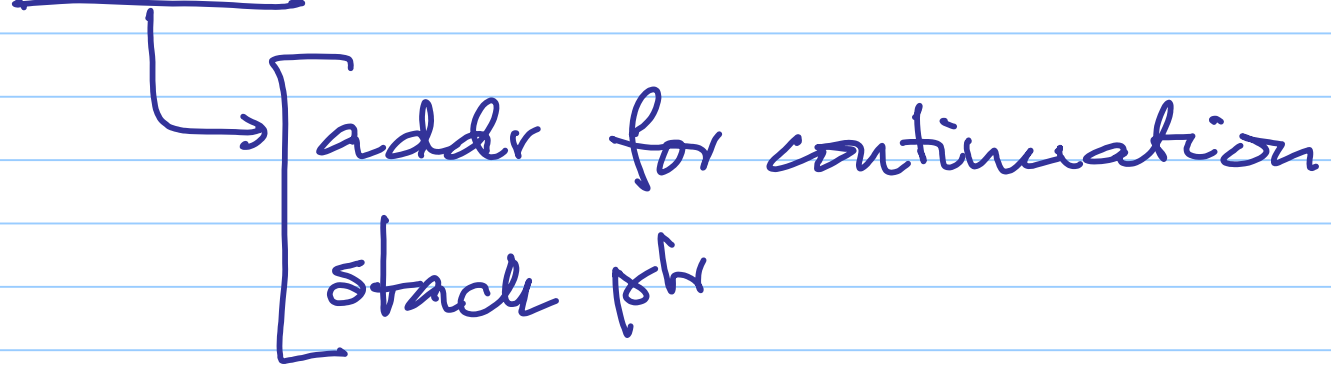
↳ 3 stacks

→ context

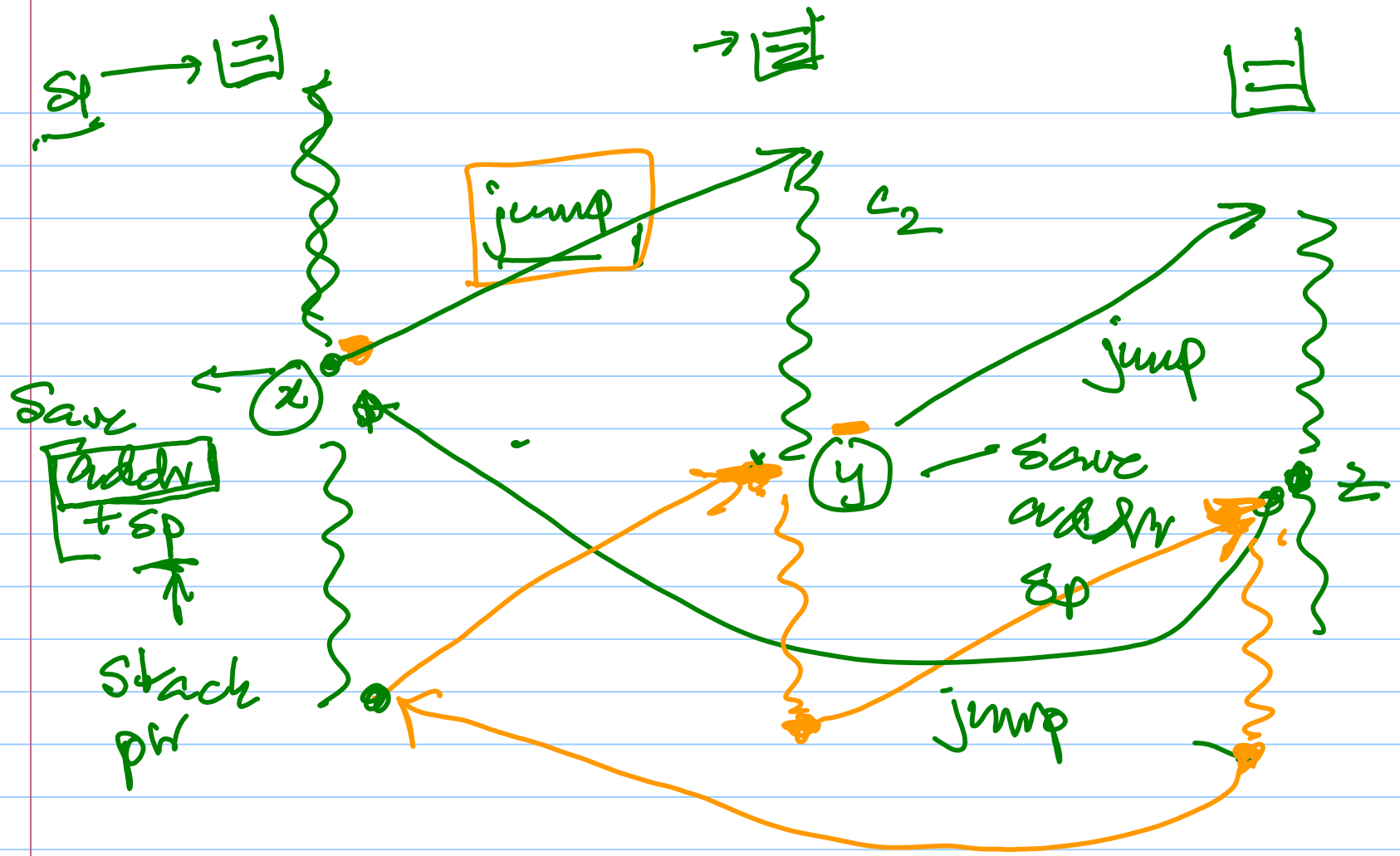
1. where did a coroutine stop ← return addr
2. what was the stack ptr

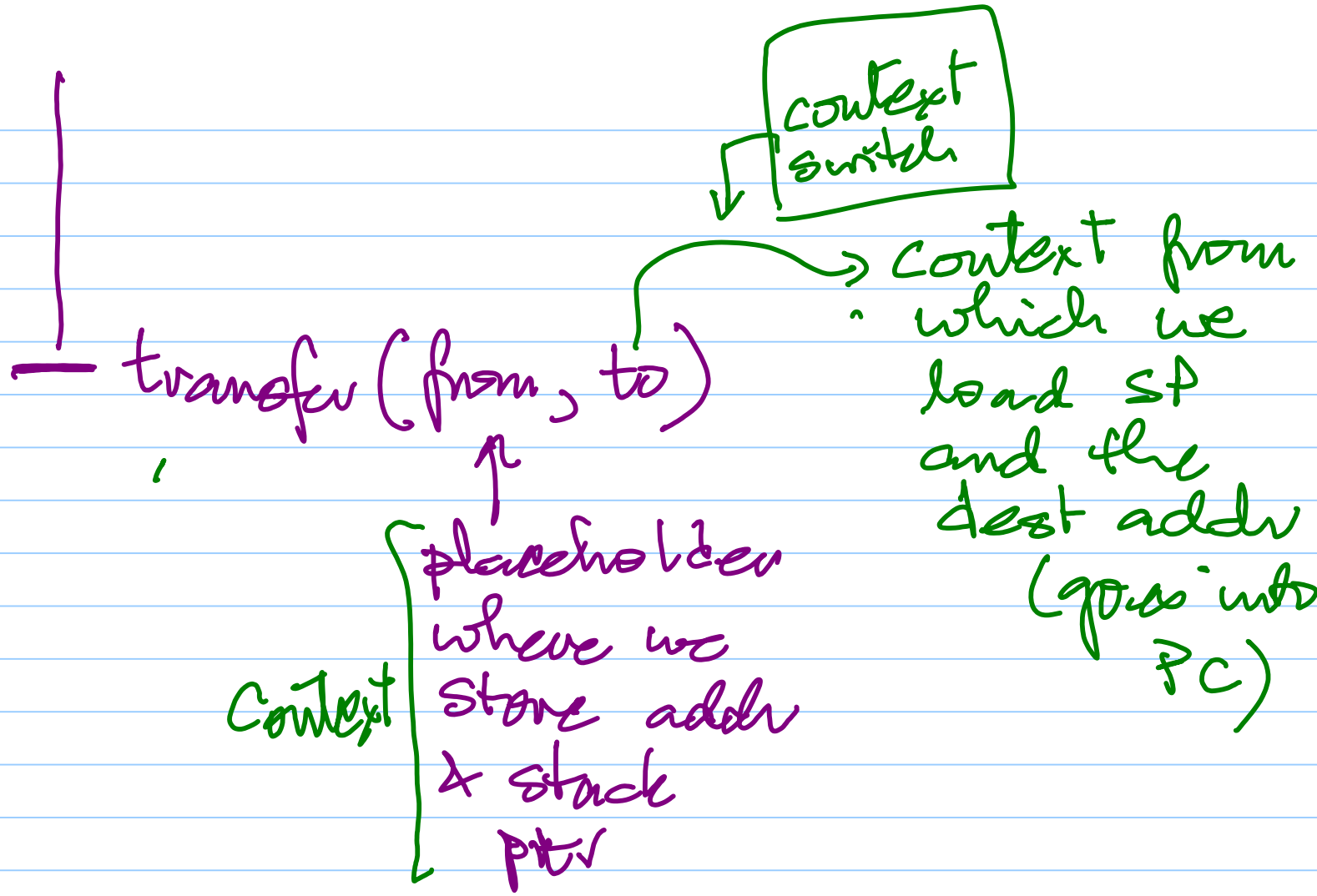
Coroutines

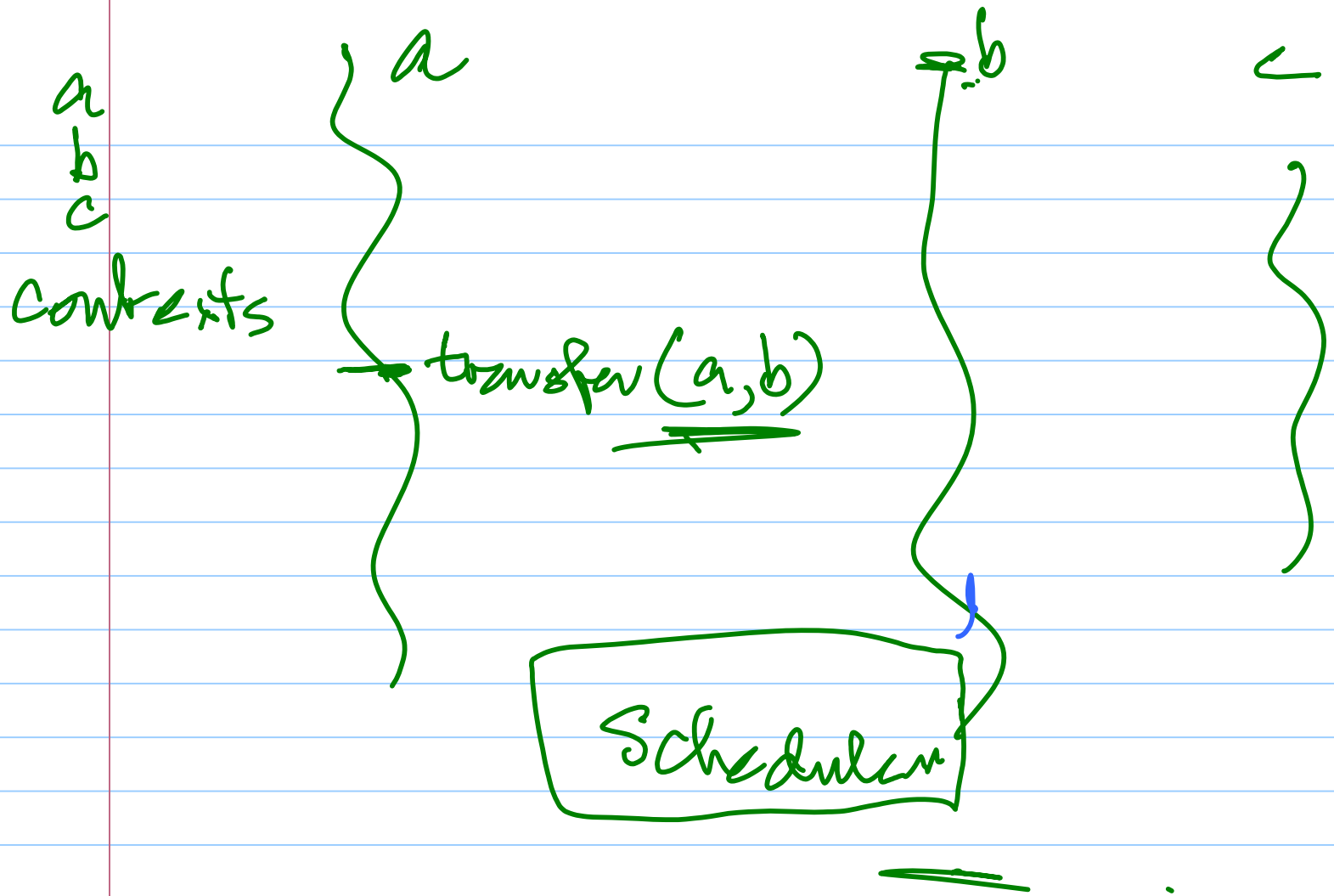
- one stack per coroutine
- one context per coroutine



multiple threads
from 1 thread


sp
jump
$c_2$
x
y
z
Save
addr
+sp
Stack
ptr
jump
save
addr
sp
jump

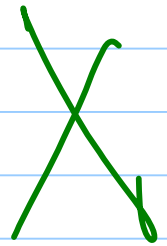

context switch
transfer (from, to)
context from which we load SP and the dest addr (goes into PC)
Context
placeholder where we store addr & stack ptr


a
b
c
contexts
a
b
c
transfer(a,b)
Scheduler

transfer(from, to)

? resume

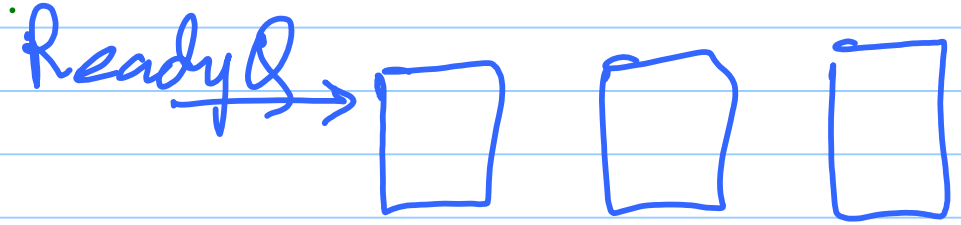
? continue



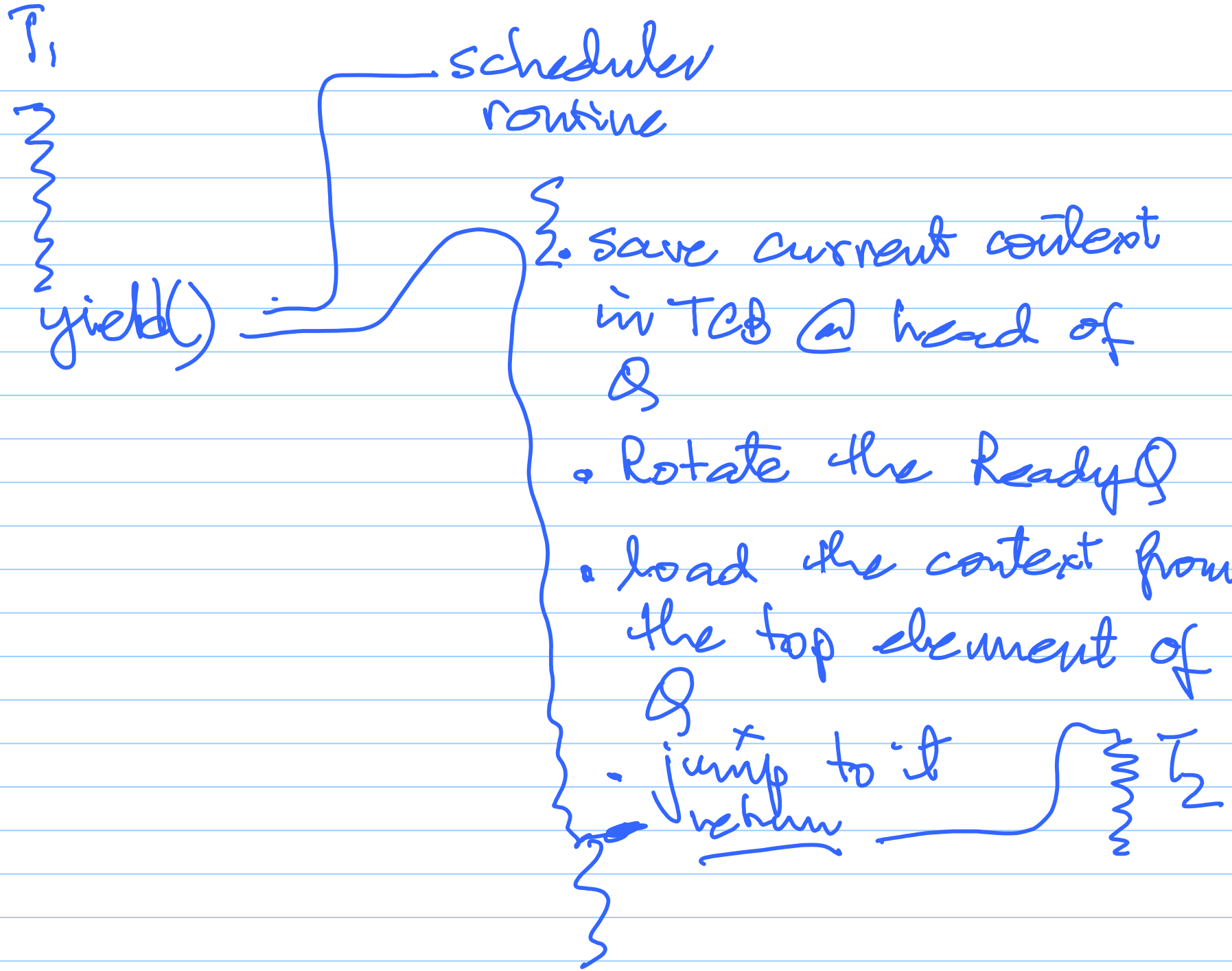
yield() ← no parameters

[ the scheduler finds from & to !!

Scheduler - data structure




main()
{ create context for $T_1$
" " " $T_2$
" " "
.
.
.
} chains them up


T1
yield()
scheduler routine
1. save current context in TCB @ head of Q
2. Rotate the ReadyQ
3. load the context from the top element of Q
4. jump to it
return
T2

f() { ∞ loop }

main()

{ create a context

↳ startthread(f)

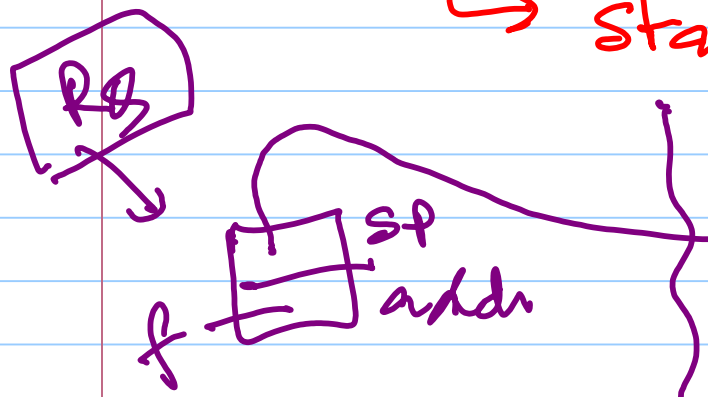
↳ allocate a stack — malloc ??

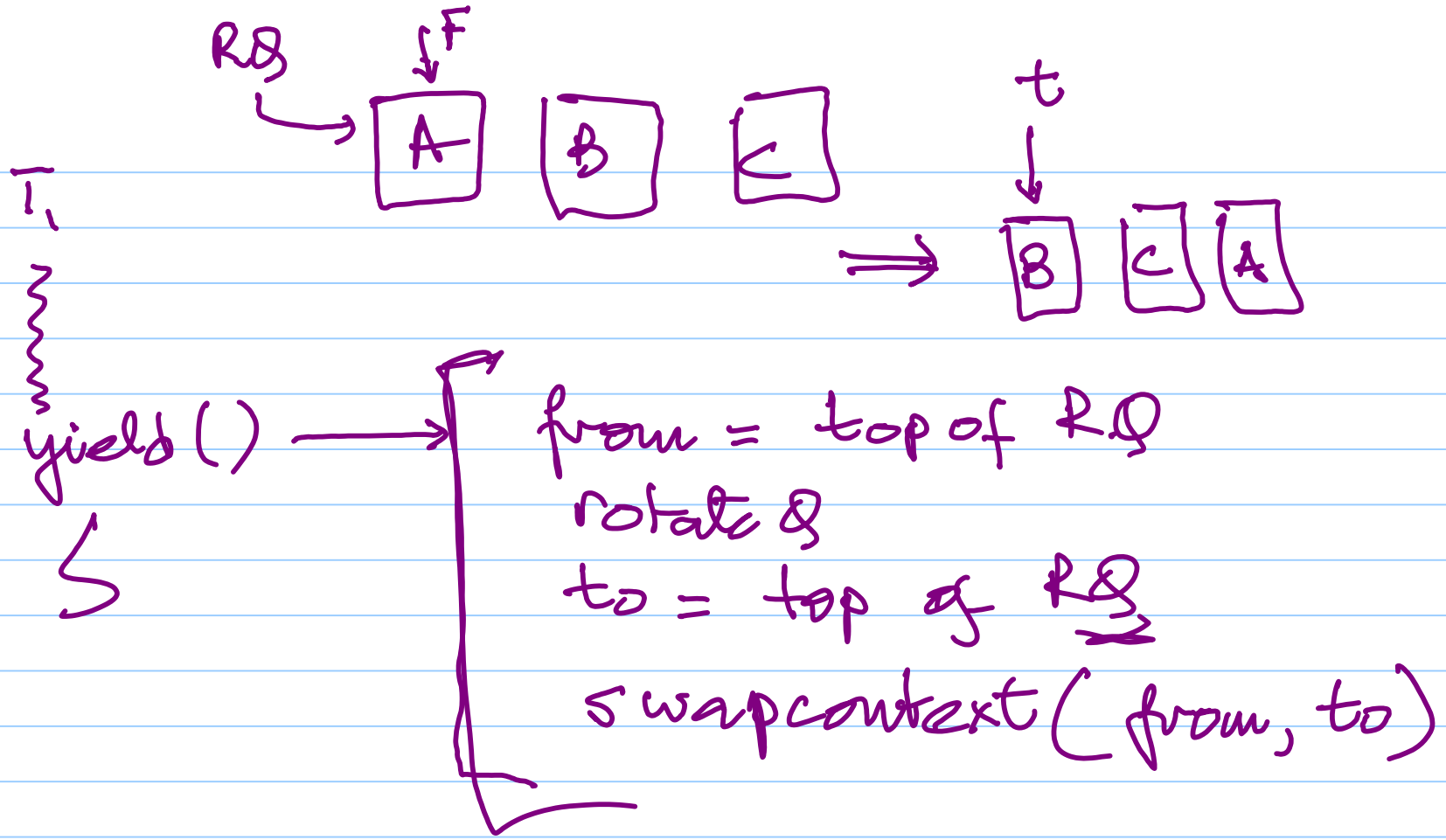
init the stack

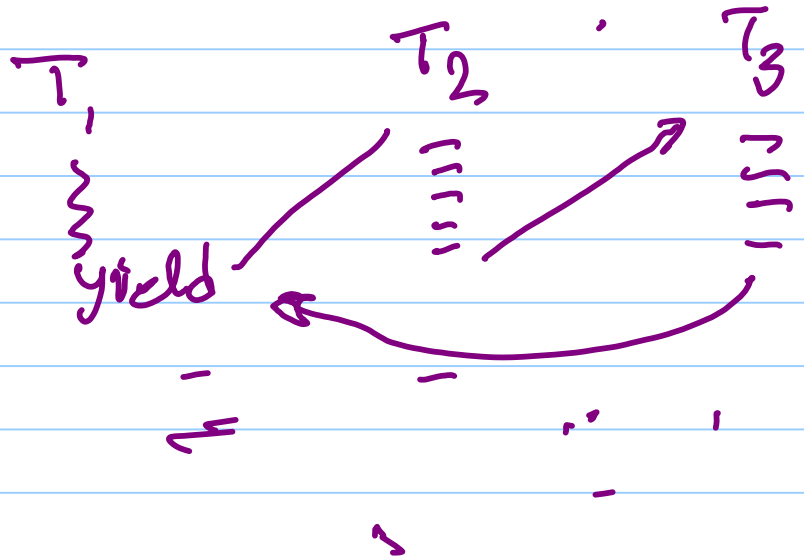
create context for (f)

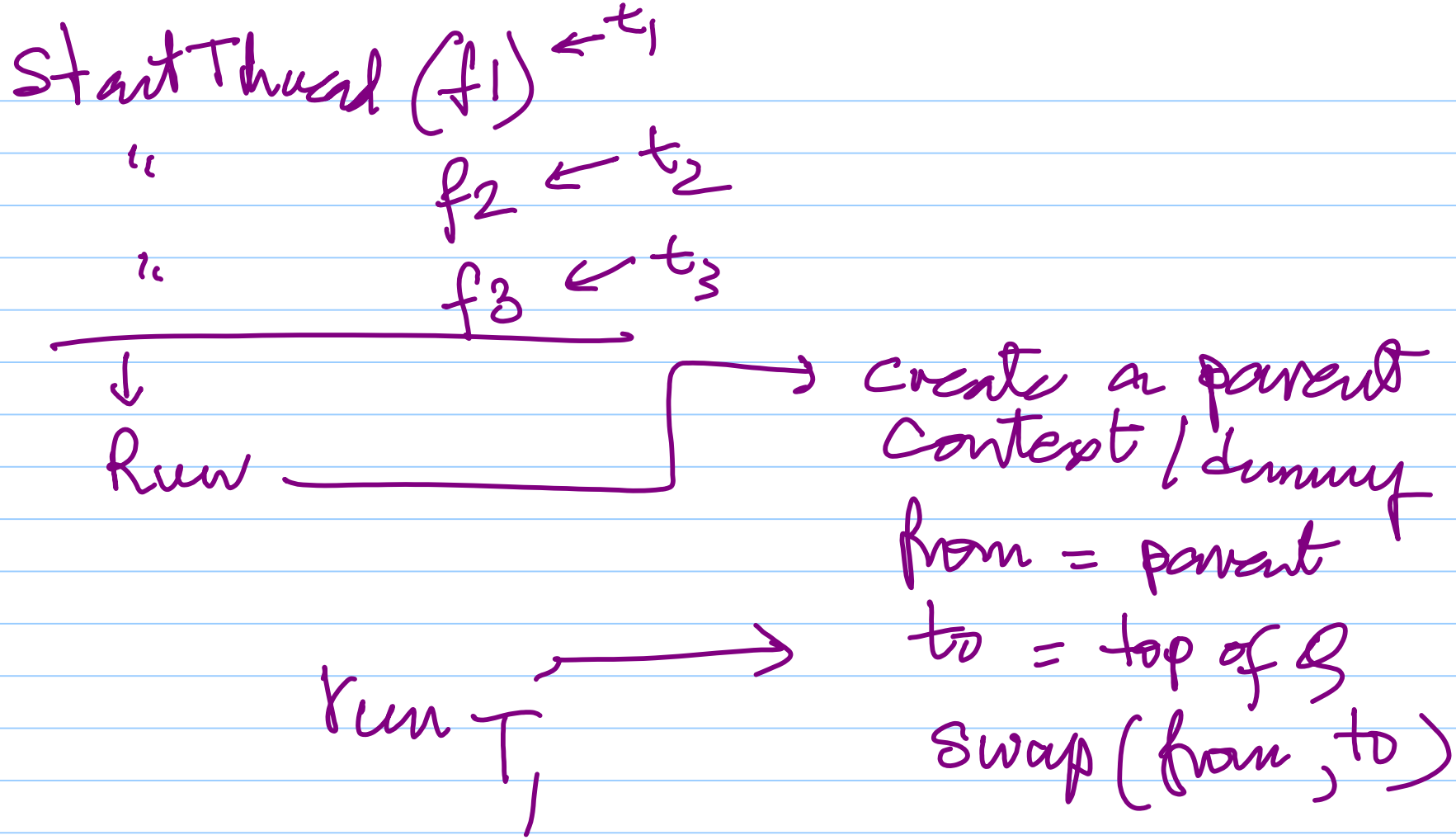
Add to Ready Q

}




RQ
F
A
B
C
t
B
C
A
T1
yield()
from = top of RQ
rotate Q
to = top of RQ
swapcontext(from, to)


T1
T2
T3
yield


StartThread (f1) ← t1
f2 ← t2
f3 ← t3
Run
create a parent context / dummy
from = parent
to = top of S
swap (from, to)
Run T1

f1() { ∞ loop  print (xxx) }

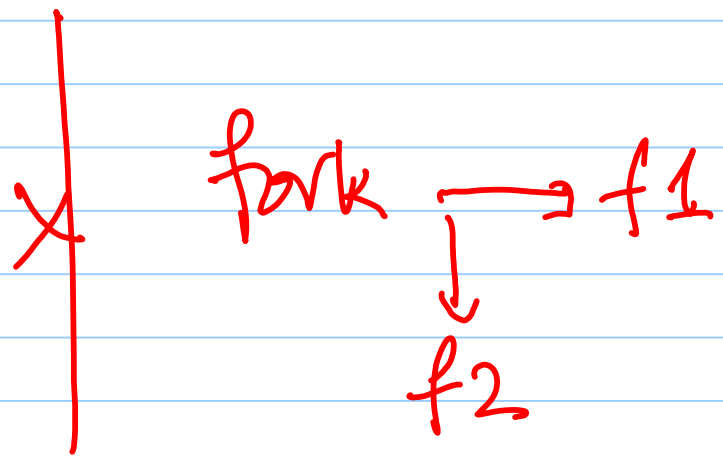
f2() { ∞ loop  print (yyy) }

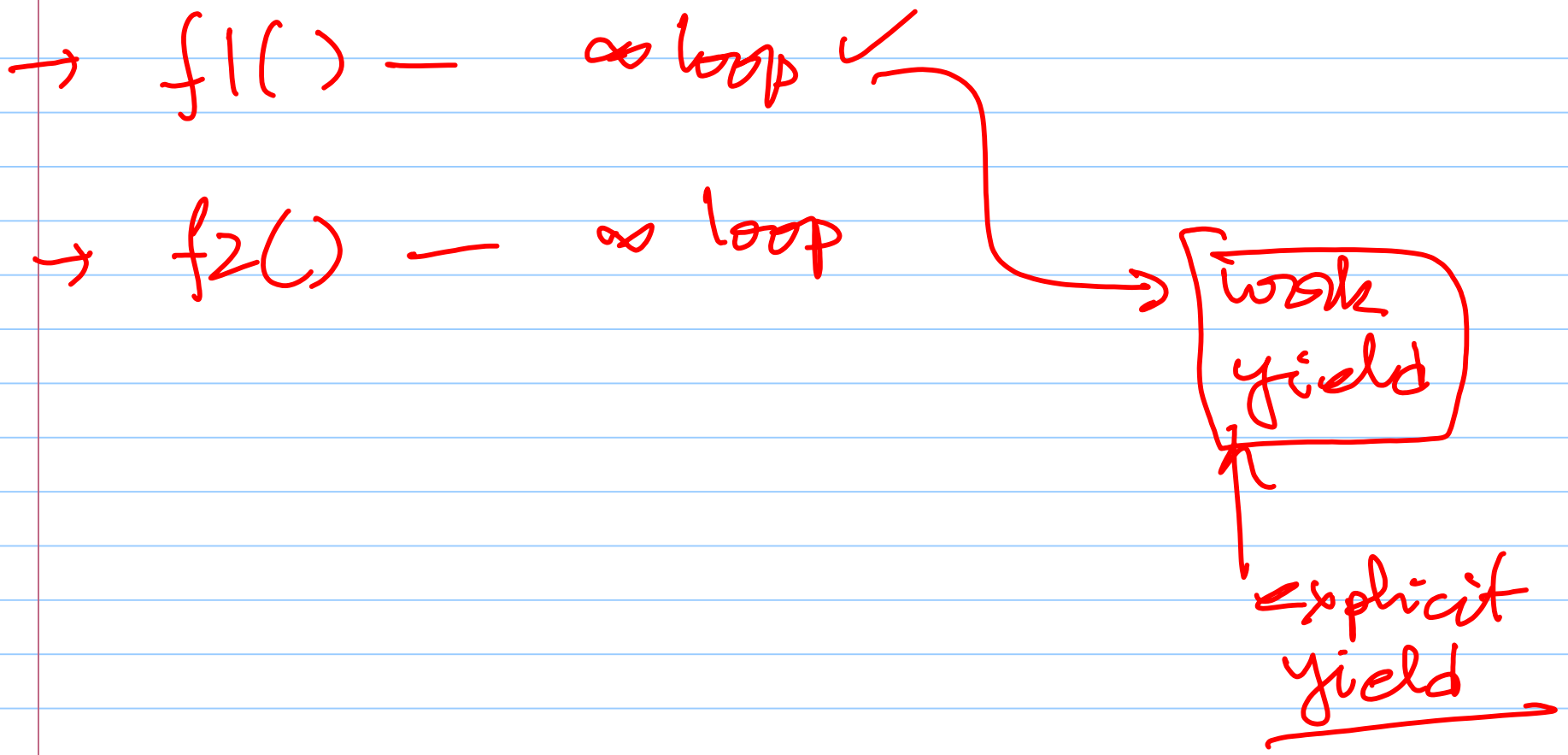
main

{ call f1

call f2

}

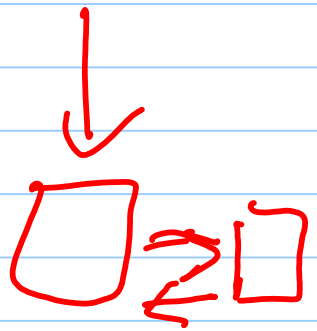


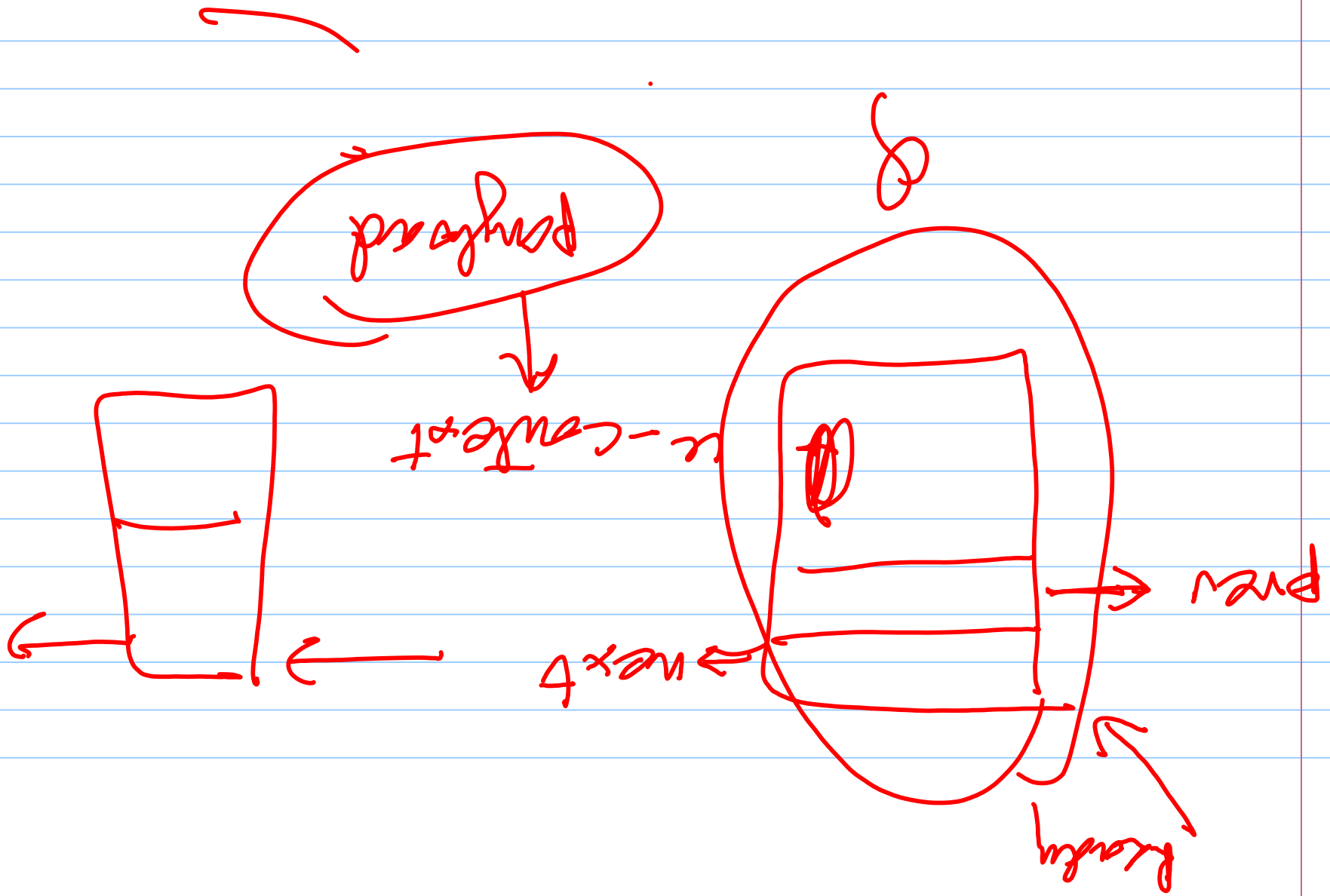

f1() — ∞ loop ✓
f2() — ∞ loop
work
yield
explicit
yield

Add (&ptr, item)

Rot (&ptr)

← Del (&ptr




payload
r_context
prev
next
header