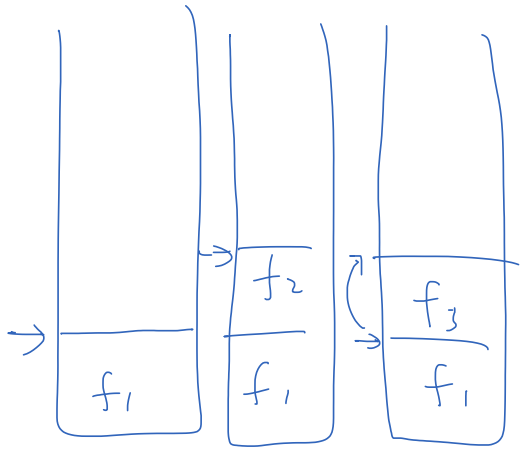
f1 printDigitPI() {

→ f2 printNextDigit()

→ f3 yield()

}



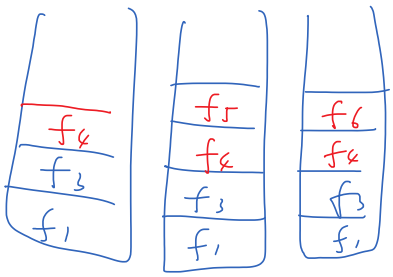
- user instructions
- no external event (int, timer)

f4 run-new-thread() {

f5 ⟷ pickNew()

f6 ⟷ switch()

}



systemCall() {

logical function code block

run-new-thread()

}

T1. producer T2. consumer

T_1: producer T_2: consumer

	T_1	T_2
✓ count	rw	rw
✓ buffer	w	r
✓ in	w	—
✓ out	—	w

not Shared

in = 0; out = 0; count = 0;

T_1

```
while (true) {
    /* produce an item and put in nextProduced */
    while (count == BUFFER_SIZE)
        ; // do nothing
    buffer [in] = nextProduced;
    in = (in + 1) % BUFFER_SIZE;
    count++;
}
```

T_2

```
while (true) {
    while (count == 0)
        ; // do nothing
    nextConsumed = buffer[out];
    out = (out + 1) % BUFFER_SIZE;
    count--;
    /* consume the item in nextConsumed
}
```

Buffer_Size = C > 0

T_2 runs first
↓
count == 0 ⇒ Busy waiting

Timer ~~fires~~ is up ← Schedule to run T_1 ⇒ counter == 0

Int → buffer[2] ← in ← 1;

Int

page 5