Embedded Operating Systems

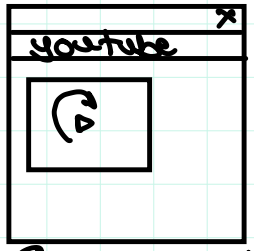
Trainer: Nilesh Ghule

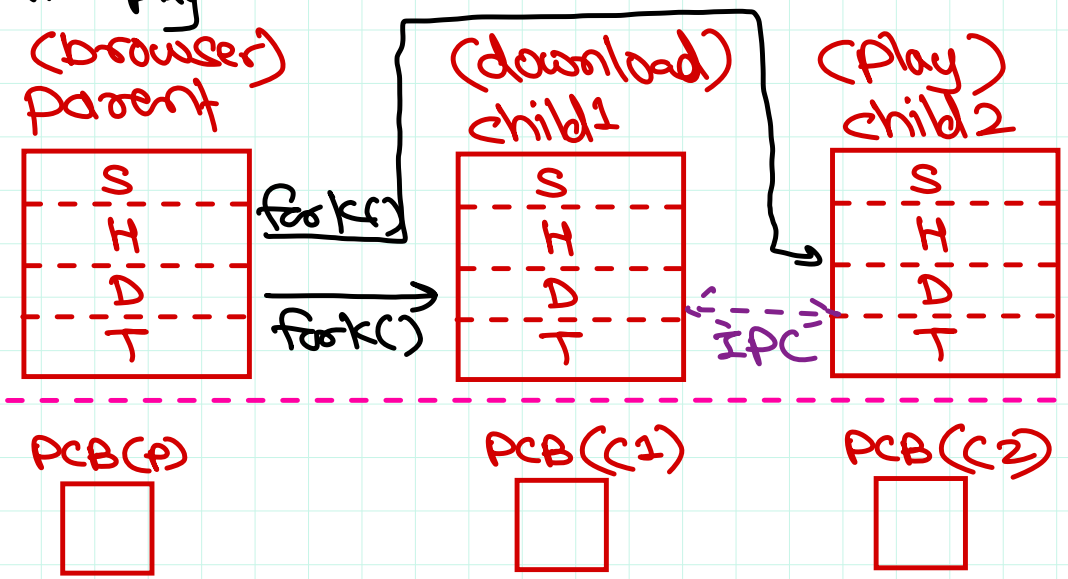
Multi-Threading

Problem: to perform multiple tasks concurrently.
a.k.a. multi-tasking

Solving: ① process based multi-tasking
- create new processes to perform multiple tasks concurrently.

② thread based multi-tasking
a.k.a. multi-threading.
- create new threads to perform multiple tasks concurrently within same process.

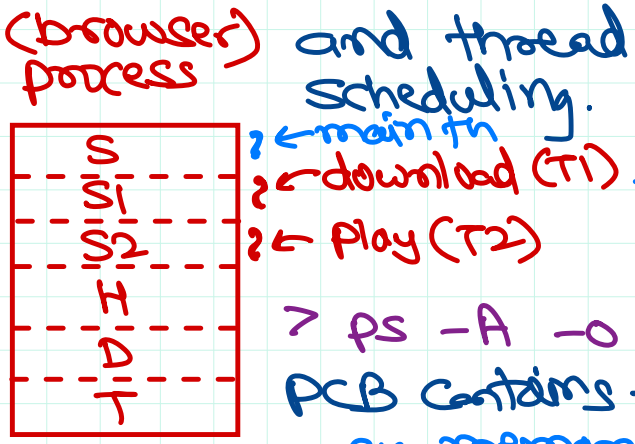




* process based multi-tasking

* Thread is a lightweight process.
ⓐ For each thread only a new stack & control block is created. Other sections are shared with parent process.
ⓑ Inter-thread comm^n is more efficient than IPC.

In modern OS, process is a container that holds resources required for exec. and thread is unit of execution scheduling.



Each process have one thread created by default called as main thread.

> ps -A -o pid,nlwp,cmd

PCB contains - info about resources.
e.g. memory (base/limit or page tbl), files (OFDT), IPC (signals), pid, ppid, exit status, ...

TCB contains - info abt exec
e.g. tid, sched info (state, time quantum), kernel stack (exec ctx)

* thread based multi-tasking

POSIX – standard for UNIX OS
- Portable Operating System Interface for X-windows(i.e. UNIX).
- Commands, syscalls, ...
- libraries: Cstd, thread lib, ...

POSIX Thread Library

Step1: implement a thread func i.e. fn to be executed by the thread.

```
void * thread_func (void * param) {
    //...
}
```

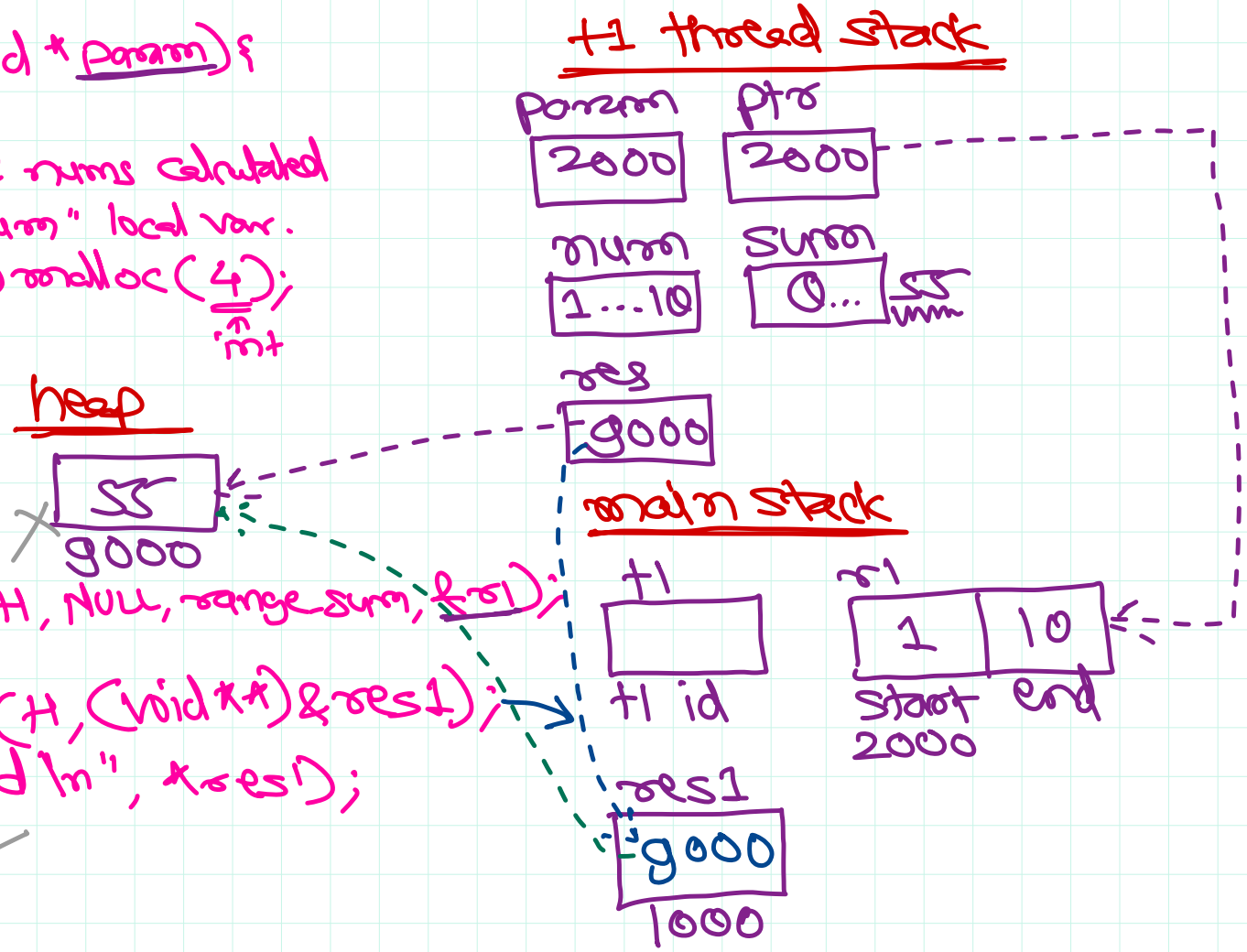
Step2: create a thread - pthread_create()

```
pthread_t tid;
ret= pthread_create (&tid, attrs, thread_func, args);
```

attributes: ① stack size ② priority ③ sched policy ④ ...
- NULL → default attrs

```
void* range_sum(void *param){
    //...
    // sum of range of nums calculated
       & kept in "sum" local var.
    int *res=(int*)malloc(4);
    *res = sum;
    return res;
}

int main(){
    //...
    pthread_create(&t1, NULL, range_sum, &r1);
    int *res1;
    pthread_join(t1,(void**)&res1);
    pf("res = %d\n", *res1);
    free(res1);
    //...
}
```



Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>