CS416 Project2 – Planning

pthread\_yield():

if NOT main:

- change current thread state from RUNNING to READY

- increment thread\_counter by 1

- call search\_next\_thread() (call it T)

- if T is null:

- Done (do nothing)

- if T is BLOCKED (means search\_next\_thread() returned a thread blocked by join):

-

- change T’s thread state to RUNNING

- change current thread ID pointer to T

- context switch from current thread to T

pthread\_join(thread T2, void \*\*value\_ptr):

- if called for first time:

- run scheduler (until T2 is done)

- done

- fetch T2 from TCB table structure

- if T2 state is DONE:

-if value\_ptr is not null:

- get T2’s return value from TCB hash table

- else:

- pthread\_yield()

pthread\_create():

- create TCB

- make the context

- push thread to data structure (ordered linked list and TCB hash table)

pthread\_exit(void \*value\_ptr):

- change pthread\_exit flag to true

- free the stack, free the context (retrieve from TCB hash table)

- store value\_ptr (thread’s return value) into return-value attribute

- change thread status to DONE

pthread\_create\_helper():

- calls the thread function

- calls pthread\_exit if pthread\_exit flag is false

search\_next\_thread(): (let T be the thread with minimum counter)

- if T is BLOCKED and joined\_on is not null:

- check if T’s joined\_on thread’s state is DONE (retrieve from TCB hash table)

- if joined\_on’s state is DONE:

- fetch joined\_on thread’s return value if any

- return T

- else:

- look at thread with next min counter

- else:

- return T