OBL2-OS

# 1.Processes and Threads

1. Threads are used for small tasks while processes are used for things like the execution of applications.
2. 1. It is desirable to use threads for parallel processing when the program wants to run several smaller tasks simultaneously that works together. Like when you want to process user input while performing other tasks at the same time and have a responsive user interface.
   2. It is desirable to use processes for parallel processing when executing larger and more heavy programs that require more power to finish.
3. The TCB contains all the Thread-specific information needed to manage the thread in question, that’s why each thread needs their own TCB
4. Voluntary is when the thread is finished with what it was doing or when the thread joins another thread.

Involuntary is when an interrupt or exception happens, or when another thread has higher priority, so the thread is forced out.

When a Voluntary thread switch happens, it saves the registers on the old stack, switches to the new stack of the new thread, restores the registers from the new stack and returns to operation.

When an involuntary thread switch happens it’s because wither a timer or I/O interrupt tells the OS some other thread should run or when an interrupt handler calls switch(). When resumed, return from handler resumes the thread or process in question. Thus processor context is saved and restored twice.

# 2. C program with POSIX threads

1. When main creates a thread it executes the \*go method which prints the line “Hello from thread ##” and exits thread with the message “Thread returned with (100+Threadnumber)”
2. The order changes each time because the order of the threads reaching the go method varies from run to run due to processing speed and what the processor is currently doing.
3. The minimum is 1(only thread 8 running due to other threads either being finished or not having started yet). The maximum is 10(if thread 8 is the last thread to start and no other thread has finished yet).
4. The pthread\_join() function waits for the thread specified by thread to terminate, unless the thread has already terminated in which case it will return immediately.
5. When it gets to thread 5 no thread will finish before 2 seconds have passed.
6. When pthread\_join() returns for a thread it means that the selected thread is either terminated or no longer set to run.