1. Поддержка многопоточности с плоской организацией планирования

   1.1В фале kern/env.c:

      Добавлены функции thread_create(), thread_destroy(), thread_free(), thread_join()

      thread_create() создаёт новый поток и возвращает его id

      thread_destroy() уничтожает поток

      thread_free() очищает и удаляет поток в конце функции

      thread_join() синхронизация потоков

   1.2 В файле lib/fork.c

      Добавлены пользовательские функции thread_create(), thread_destroy(), thread_free(), thread_join() на основе ядерных функций с помощью вызова функции syscall()

2. Синхронизация потоков с помощью мьютексов

   mutex_init(): Инициализация мьютекса: выделяется свободная страница и устанавливаются нулевые значения.

   mutex_destroy(): Удаление мьютекса: все ожидающие потоки удаляются из очереди и устанавливаются в RUNNABLE.

   mutex_unlock(): Разблокировка мьютекса - изменение заблокированного значения на 0. Если лист ожидания не пустой, env запускается по порядку, он устанавливается как владелец потока и этот поток устанавливается как исполняемый.

   mutex_lock(): Функция блокирует мьютекс. Значение "Locked" изменяется атомарной операцией xchg. Если возвращается 0 и никто не ждет мьютекс, мы устанавливаем владельца в текущий env. Иначе, env добавляется в конец списка и устанавливается в NOT RUNNABLE.