Thread implementation in PHP

A thread of execution is the smallest sequence of programmed instructions that can be managed independently by a scheduler, which is typically a part of the operating system. In the PHP, you can divide and conquer your planned process efficiently by Thread.

The below is the simple example of thread:

```
<?php

class workerThread extends Thread
{
        public function __construct($i)
        {
                  $this->i=$i;
        }

        public function run()
        {
                while(true)
                {
                        echo $this->i;
                        sleep(1);
                }
        }
}

for($i=0;$i<50;$i++)
{
        $workers[$i]=new workerThread($i);
        $workers[$i]->start();
}

?>
```

To understand of the concept for Thread, below example will be super helpful:

```php
<?php
class STD extends Thread{
    public function put()
        {
        $this->synchronized(function()
                {
                   for($i=0;$i<7;$i++)
                        {
                            printf("%d\n",$i);
                            $this->notify();
                            if($i < 6)
                                    $this->wait();
                            else
                                exit();
                            sleep(1);
                        }
        });

    }

        public function flush()
        {
                $this->synchronized(function()
                {
            for($i=0;$i<7;$i++)
                        {
                            flush();
                            $this->notify();
                            if($i < 6)
                                    $this->wait();
                            else
                                exit();
                    }
                });
        }
}

class A extends Thread
{
    private $std;
    public function __construct($std)
        {
        $this->std = $std;
    }
    public function run()
        {
        $this->std->put();
    }
}

class B extends Thread
{
    private $std;
    public function __construct($std)
        {
        $this->std = $std;
    }
    public function run()
        {
        $this->std->flush();
    }
}

ob_end_clean();
echo str_repeat(" ", 1024);
$std = new STD();
$ta = new A($std);
$tb = new B($std);
$ta->start();
$tb->start();
```