EECS 268 Lab 2: CPU Scheduler

FIFO

(Queue)

CPU Scheduler

Process 1 → Process 2 → Process 3 …

Each process contains a call stack

LIFO Process

assuming you have node.py

linked\_stack.py

linked\_queue.py

from linked\_stack import Stack

class Process:

def \_\_init\_\_(self,id):

self.id = id

self. stack = Stack()

self.stack.push(‘main’)

def call(self,fn):

return self.stack.push(fn)

def end(self):

return self.stack.pop()

from process import Process

from linked\_queue import Queue

class Scheduler:

def \_\_init\_\_(self, cmds):

self.queue = Queue()

self.cmds = cmds

def start(self,process):

self.queue.enqueue(Process(process))

print(f’{process} added to queue)

def call(self,func):

try:

process = self.queue.peek()

except: RuntimeError as err:

print(“Could not call process”)

else:

process call(func)

self.queue.dequeue()

self.enqueue(process)

print(f’{process.id} calls {func}’)

try:

process = self.queue.peek()

except:

runtime error

else:

func = process.end()

process.stack.pop()

print(f’{process.id} returns from {func}’)

if func == ‘main’:

self.queue.dequeue()

print(f’{process.id} has ended’)

else:

self. queue.dequeue()

self.queue.enqueue()