logo, grafik, ekran görüntüsü, grafik tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Enes Smajli – b201210590-1.A*

*Ali Eren Yiğit – b201210062-1.A*

*Yasin Gökçe – b201210010-1.C*

*Mahi Mohamad AbdulHakim – b201210605-1.A*

*M. Yusuf Allahverdi – b2012210032-1.A*

İşeltim sistemleri Proje ödevi

45.Grup

* Projenin kısımları :
* DoublyLinkedList :
* Düğüm sınıfı kullanılarak çift bağlı liste oluşturuldu.

Kullanılan metotlar ;

\*addLast

\*arayaEkle

\*deleteNode

\*olumDegeriArttir

\*IsEmpty

\*sonraki

* Node :
* Düğüm sınıfı ve bu düğüme ait nesne ve metotlar oluşturuldu.

Kullanılan metotlar ;

\*Node (Kurucu metod)

\*delete

* Process :
* İd, varişzamanı, oncelik, kalanzaman, printers, color … özelliklerine ve birkaç metoda ait Process sınıfı oluşturdu.

Kullanılan metotlar ;

\*Process (Kurucu metod)

\*getId

\*yeniBasladi

\*bitti

\*yurutuluyor

\*zamanAsimi

\*askida

\*getStatus

* Dosya :

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

* main:

package OSProjectProva1;

public class mainProva1 {

public static void main(String[] args) {

DoublyLinkedList tumListe = new DoublyLinkedList();

Dosya dosya = new Dosya();

DoublyLinkedList p0 = new DoublyLinkedList();

DoublyLinkedList p1 = new DoublyLinkedList();

DoublyLinkedList p2 = new DoublyLinkedList();

DoublyLinkedList p3 = new DoublyLinkedList();

Node node;

Node nodeTutucu;

System.out.println("Pid varış öncelik cpu MBytes prn scn modem cd status");

int oncelik;

tumListe = dosya.Oku("C:\\Users\\Enes\\Desktop\\donem 4\\PDP\\Eclipse\\OSProject\\girisTest.txt");

// onceliklerine göre listelere atılıyor...

for (int i = 0; i < tumListe.size; i++) {

oncelik = tumListe.head.process.oncelik;

if (oncelik == 0) {

node = new Node(tumListe.head.process);

p0.addLast(node);

}

if (oncelik == 1) {

node = new Node(tumListe.head.process);

p1.addLast(node);

}

if (oncelik == 2) {

node = new Node(tumListe.head.process);

p2.addLast(node);

}

if (oncelik == 3) {

node = new Node(tumListe.head.process);

p3.addLast(node);

}

tumListe.sonraki();

}

int yenimiBasladi = 0;

int calistigiSure = 0;

int calisacagiSure = 0;

int gercekZaman = 0;

int gercekZamanIcerideArtmadi = 0;

while (true) {

gercekZamanIcerideArtmadi = 0;

if (p0.head != null) {

if (p0.head.process.varisZamani <= gercekZaman) {

calisacagiSure = p0.head.process.calisacagiSure;

calistigiSure = p0.head.process.calistigiSure;

yenimiBasladi = calisacagiSure - calistigiSure;

if (yenimiBasladi == calisacagiSure) {

p0.head.process.yeniBasladi(gercekZaman, yenimiBasladi);

p0.head.process.calistigiSure++;

gercekZaman++;

} else if (yenimiBasladi < calisacagiSure && yenimiBasladi > 0) {

p0.head.process.yurutuluyor(gercekZaman, yenimiBasladi);

p0.head.process.calistigiSure++;

gercekZaman++;

}

calisacagiSure = p0.head.process.calisacagiSure;

calistigiSure = p0.head.process.calistigiSure;

yenimiBasladi = calisacagiSure - calistigiSure;

if (yenimiBasladi == 0) {

p0.head.process.bitti(gercekZaman, yenimiBasladi);

p0.sonraki();

}

p0.olumDegeriArttir(p0, p1, p2, p3, gercekZaman);

continue;

} else

gercekZamanIcerideArtmadi++;

}

if (p1.head != null) {

if (p1.head.process.varisZamani <= gercekZaman) {

calisacagiSure = p1.head.process.calisacagiSure;

calistigiSure = p1.head.process.calistigiSure;

yenimiBasladi = calisacagiSure - calistigiSure;

if (yenimiBasladi == calisacagiSure) {

p1.head.process.yeniBasladi(gercekZaman, yenimiBasladi);

p1.head.process.calistigiSure++;

gercekZaman++;

calisacagiSure = p1.head.process.calisacagiSure;

calistigiSure = p1.head.process.calistigiSure;

yenimiBasladi = calisacagiSure - calistigiSure;

if (yenimiBasladi == 0) {

p1.head.process.bitti(gercekZaman, yenimiBasladi);

p1.sonraki();

} else {

p1.head.process.oncelik = 2;

p1.head.process.askida(gercekZaman, yenimiBasladi);

nodeTutucu = new Node(p1.head.process);

p2.arayaEkle(nodeTutucu, gercekZaman);

p1.sonraki();

}

}

p0.olumDegeriArttir(p0, p1, p2, p3, gercekZaman);

continue;

} else

gercekZamanIcerideArtmadi++;

}

if (p2.head != null) {

if (p2.head.process.varisZamani <= gercekZaman) {

calisacagiSure = p2.head.process.calisacagiSure;

calistigiSure = p2.head.process.calistigiSure;

yenimiBasladi = calisacagiSure - calistigiSure;

if (yenimiBasladi == calisacagiSure) {

p2.head.process.yeniBasladi(gercekZaman, yenimiBasladi);

p2.head.process.calistigiSure++;

gercekZaman++;

calisacagiSure = p2.head.process.calisacagiSure;

calistigiSure = p2.head.process.calistigiSure;

yenimiBasladi = calisacagiSure - calistigiSure;

if (yenimiBasladi == 0) {

p2.head.process.bitti(gercekZaman, yenimiBasladi);

p2.sonraki();

} else {

p2.head.process.oncelik = 3;

p2.head.process.askida(gercekZaman, yenimiBasladi);

nodeTutucu = new Node(p2.head.process);

p3.arayaEkle(nodeTutucu, gercekZaman);

p2.sonraki();

}

} else if ((calisacagiSure > calistigiSure)) {

p2.head.process.yurutuluyor(gercekZaman, yenimiBasladi);

p2.head.process.calistigiSure++;

gercekZaman++;

calisacagiSure = p2.head.process.calisacagiSure;

calistigiSure = p2.head.process.calistigiSure;

yenimiBasladi = calisacagiSure - calistigiSure;

if (yenimiBasladi == 0) {

p2.head.process.bitti(gercekZaman, yenimiBasladi);

p2.sonraki();

} else {

p2.head.process.oncelik = 3;

p2.head.process.askida(gercekZaman, yenimiBasladi);

nodeTutucu = new Node(p2.head.process);

p3.arayaEkle(nodeTutucu, gercekZaman);

p2.sonraki();

}

}

p0.olumDegeriArttir(p0, p1, p2, p3, gercekZaman);

continue;

} else

gercekZamanIcerideArtmadi++;

}

if (p3.head != null) {

if (p3.head.process.varisZamani <= gercekZaman) {

calisacagiSure = p3.head.process.calisacagiSure;

calistigiSure = p3.head.process.calistigiSure;

yenimiBasladi = calisacagiSure - calistigiSure;

if (yenimiBasladi == calisacagiSure) {

p3.head.process.yeniBasladi(gercekZaman, yenimiBasladi);

p3.head.process.calistigiSure++;

gercekZaman++;

calisacagiSure = p3.head.process.calisacagiSure;

calistigiSure = p3.head.process.calistigiSure;

yenimiBasladi = calisacagiSure - calistigiSure;

if (yenimiBasladi == 0) {

p3.head.process.bitti(gercekZaman, yenimiBasladi);

p3.deleteNode(p3.head);

} else {

p3.head.process.askida(gercekZaman, yenimiBasladi);

nodeTutucu = new Node(p3.head.process);

p3.arayaEkle(nodeTutucu, gercekZaman);

p3.deleteNode(p3.head);

}

} else if (yenimiBasladi < calisacagiSure && yenimiBasladi > 0) {

p3.head.process.yurutuluyor(gercekZaman, yenimiBasladi);

p3.head.process.calistigiSure++;

gercekZaman++;

calisacagiSure = p3.head.process.calisacagiSure;

calistigiSure = p3.head.process.calistigiSure;

yenimiBasladi = calisacagiSure - calistigiSure;

if (yenimiBasladi == 0) {

p3.head.process.bitti(gercekZaman, yenimiBasladi);

p3.deleteNode(p3.head);

} else {

p3.head.process.askida(gercekZaman, yenimiBasladi);

nodeTutucu = new Node(p3.head.process);

p3.arayaEkle(nodeTutucu, gercekZaman);

p3.deleteNode(p3.head);

} //dugumu sona tasıyıp her şeyi bir sola kaydıracağız.

}

p0.olumDegeriArttir(p0, p1, p2, p3, gercekZaman);

continue;

} else

gercekZamanIcerideArtmadi++;

}

if (gercekZamanIcerideArtmadi == 4) {

p0.olumDegeriArttir(p0, p1, p2, p3, gercekZaman);

gercekZaman++;

}

if (p0.IsEmpty() && p1.IsEmpty() && p2.IsEmpty() && p3.IsEmpty())

break;

}

}

}

* Ekran çıktısı :

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu