Contents

Raymonds Algorithm - A Centralized Approach

Sabbir Rashid & Rob Berman
Rensselaer Polytechnic Institute

I ABSTRACT

II INTRODUCTION

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

II.I Problem Statement

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

III METHODOLOGY

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

III.I Raymond's Functions

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

III.I.i assignToken(Process p)

```
public void assignToken(Process p) {
  if ( (p.holderEnum == Process.HolderEnum.Self
      ) && (!p.usingResource) && (!p.
      requestQueue.isEmpty()) ) {
```

```
    holderProc = p.requestQueue.pop() ;

    if (p.getProcessID() == holderProc.
        getProcessID()) { //i.e. the process p is at
         the front of its own queue
      p.holderEnum = Process.HolderEnum.Self;
    } else {
      p.holderEnum = Process.HolderEnum.
          Neighbor;
      holderProc.holderEnum = Process.
          HolderEnum.Self ;
    }

    p.asked = false;

    if (p.holderEnum == Process.HolderEnum.Self
        ) {
      p.usingResource = true;
    } else {
      assignToken(holderProc); // Check this,
          supposed to be "send token to holder"
    }
  }
}
```

III.I.ii sendRequest(Process p)

```
public void sendRequest(Process p) {
  if ( (p.holderEnum != Process.HolderEnum.Self)
      && (!p.requestQueue.isEmpty()) && (!p.
      asked) ) {
    sendRequest(holderProc);
    p.asked = true;
  }
}
```

III.I.iii requestResource(Process p)

```
public void requestResource(Process p) {
  p.requestQueue.push(p);
  assignToken(p);
  sendRequest(p);
}
```

III.I.iv releaseResource(Process p)

```
public void releaseResource(Process p) {
  p.usingResource = false;
  assignToken(p);
  sendRequest(p);
}
```

III.I.v receivedRequestFromNeighbor(Process p, Process neighbor)

```
public void receivedRequestFromNeighbor(Process
    p, Process neighbor) {
  p.requestQueue.push(neighbor);
  assignToken(p);
  sendRequest(p);
}
```

III.I.vi receivedToken(Process p)

```
public void receivedToken(Process p) {
  p.holderEnum = Process.HolderEnum.Self ;
  holderProc = p;
  assignToken(p);
  sendRequest(p);
}
```

IV CONCLUSIONS

Quisque consectetuer. In suscipit mauris a dolor pellentesque consectetuer. Mauris convallis neque non erat. In lacinia. Pellentesque leo eros, sagittis quis, fermentum quis, tincidunt ut, sapien. Maecenas sem. Curabitur eros odio, interdum eu, feugiat eu, porta ac, nisl. Curabitur nunc. Etiam fermentum convallis velit. Pellentesque laoreet lacus. Quisque sed elit. Nam quis tellus. Aliquam tellus arcu, adipiscing non, tincidunt eleifend, adipiscing quis, augue. Vivamus elementum placerat enim. Suspendisse ut tortor. Integer faucibus adipiscing felis. Aenean consectetuer mattis lectus. Morbi malesuada faucibus dolor. Nam lacus. Etiam arcu libero, malesuada vitae, aliquam vitae, blandit tristique, nisl.