gitlog.tex)

Project 2

by Sinan Topkaya

CS 444 - Spring 2016

Abstract: This is a paper that answers project 2 questions on how to build Shortest Seek Time

WRITE-UP

Project 2

SSTF Algorithm

First of all I will be examining SSTF algorithm, I will be implementing Look algorithm. In LOOK Scheduling the arm goes only as far as final request in each direction, then it reverses direction immediately without going all the way to the end of the disk. Main idea is to sort segments increasinglt on their sizes, schedule on senders one by one. Also on a single sender, we have to select a segment that has shortest transmission time and can arrive on time. Then we have to remove the scheduled segment and repeat above step until no more segments can be scheduled on that sender. We have to schedule the next sender in the same way then stop when all segments are scheduled or when no more bandwidth left.

The idea is that it is goign to be similar to noop so I have definitely had help looking at its code. A schedule for each sender Q1, Q2, ... Qm:

- Let Qm = NULL where m = 1,2, ... , Qm
- Let N consists of all remaining segments
- Sort segments increasingly in N on segment size
- For m=1 to M
- let t=0
- foreach segment n in N
- if an,m =1 and t + sn/bm ¡ dn
- add segment n to Qm
- remove segment n from N
- let t = t +sn/bm
- Return Q1, Q2, ... , Qm

Version Control Log

Detail	Author	Description
e0f6bfa	topkayas	first commit
6443ef7	topkayas	First Commit Kernel v3.14.26
44dc181	topkayas	Summary Week 1
040e0b0	topkayas	updates
b489bd0	topkayas	update
12818e6	topkayas	update
d30ab81	topkayas	week 2 summary
18c007a	topkayas	IEEEtran
9c9e7bf	topkayas	summaries done
f4d7951	topkayas	title page
da69ee0	topkayas	summary
47b8240	topkayas	last changes
74dbc03	topkayas	1 column
902be28	topkayas	one column
eaba13e	topkayas	changes
8937dde	topkayas	changes
cbbc8ff	topkayas	starting with the pdf file
77d005b	topkayas	pdf
99459b0	topkayas	change pdf
c1e5297	topkayas	update
3eefb8f	topkayas	update
ff63e14	topkayas	update
9003cf9	topkayas	update
91630e4	topkayas	update latex
ce09049	topkayas	update latex
dcf140f	topkayas	update latex
c044b56	topkayas	homework finished
96bbd6a	topkayas	Final Fixes
b402199	topkayas	Done
10138f2	topkayas	add the connection
7fd097b	topkayas	add week 3 summary
6e91bc8	topkayas	update
da7b97a	topkayas	update
ca524ab	topkayas	update
20836ef	topkayas	Starting with the Writing Assignment 1
810ddfd	topkayas	check if everything looks fine
fba1b6f	topkayas	Checking the updated version

Reflection: 1. I think the main point was to learn about different types of schedulers and how they sort the segments and execute them.

2. I have watched many sstf and look algorith videos and spent alot of time trying to udnerstand its pseudo codes. Once that was done, the noop shceduler file which is located in the block file was really easy to understand. So I started changign the noob to sstf.

3. Approaching the problem piece by piece made it much easier. What was really frustrating was that I had to keep building the OS over and over again whenever I wanted to test it, so it took alot of time. I made sure that the scheduler cat the right scheduler and started first printing the output into the console, later I added more code to put that input into a file.

4. Creating a scheduler really made me understand how the segments work and how the scheduler procceses these segments. I learned that different schedulers have different advantages and disadventages

work log

When	What
4/26/2016	Started with Project 2, I read more about sstf-look then once that was done I made a copy of the noob scheduler and tried to implement my own sstf.
4/27/2016	I have created two lists and implemented look dispatch function which executes the next segment in icnreasing order if it cant then it goes to the segment that is lower than the current head position.
4/28/2016	Everything finally worked well with qemu and Project 2 is completely finished.