CS3224 Midterm

Intro to Operating Systems

Prof. G. Sandoval

March 28 - 30, 2022

Questions	Points	Score
1	50	
2	10	
3	10	
4	10	
5	10	
6	10	

Exam Rules:

1. The exam must represent your work alone. You may not misrepresent someone else's work as yours. You may not submit work of which you are not the sole author.
2. While working on the exam you may not communicate with any other student for any reason. This prohibition includes the use of electronic communication, such as email, texting and phone. As an exception to this rule, if you have questions during the exam, you may ask the instructors or TAs. You may not discuss the content of the exam with other students. You may not ask questions on Campuswire with public posts.
3. Do not copy or distribute the exam document or your answers at any time. After you have finished and have pushed your exam to Github, delete the exam and your answers from your computer.
4. If we suspect any academic dishonesty, the Professor will make a Zoom appointment with you right after the exam. You will need to support your answers.

Upon completion of the exam, fill in this pledge of academic honesty on Anubis, otherwise, your exam may not be graded.

I, <your name>, affirm that I have completed the exam completely on my own, without consulting with other classmates. I have followed the required rules. I understand that violating these rules would represent academic dishonesty.

Accepting the Assignment and Setting Up a Repo (Mandatory Step)

1. Head to **https://anubis.osiris.services/**
2. Press “LogIn” on the side bar, and log in with your NYU ID
3. Enter GitHub username (If not done yet)
4. Go to “Courses”, and press “Join Class” with code **iloveos** to be registered to the class
5. Go to “Assignments”, and accept the assignment
    a. Press “Create Assignment Repo” (This will create a repository on Github)

Working on the Assignment

For this assignment we will offer you **two options** for a platform to work on, or you can use your own platform. It’s up to you what you use. It’s **your responsibility** to turn it on time no matter what platform you choose. **To make this 100% clear**. Whether you choose **Anubis or Vagrant you are the only one responsible for turning it on time.** If Anubis or Vagrant doesn’t work for you 5 minutes, 5 hours or 5 days before the deadline, it’s not the Professor's responsibility or the TAs responsibility, it's **your responsibility.**

Using Anubis IDE (Option 1)

1. Select “Anubis Cloud IDE” from the Assignment
2. On the top menu bar, select “Terminal” and “New Terminal” to bring up a terminal
3. Complete your assignment in the project folder, and commit your changes for submission/grading

Notes:

- With auto save on, the IDE automatically commits your changes to GitHub every couple of minutes. Or type “autosave” in the shell to save your changes
    - Even with this option on, you are still **responsible** for saving your changes
- If you have any questions, don’t hesitate to post on Forum or email any of the TAs

Using Vagrant (Option 2)

1. Follow the instructions in the repo: **https://github.com/wabscale/cs3224-vm**

2. Pull the assignment repository from Github

3. Complete your assignment in the project folder, and commit your changes for submission/grading

Using your own environment (Option 3)

1. As always you are free to use your own linux environment
    a. Assignments are graded based on the Anubis IDE environment
2. Pull the assignment repository from Github
3. Complete your assignment in the project folder, and commit your changes for submission/grading

You must accept the assignment using Anubis for any of the options listed above.

Questions 1 should be answered in the XV6 project folder and pushed to Github.

Questions 2 to 6, and Pledge are to be submitted on Anubis like HW2’s short answer questions.

The questions can be accessed by clicking the assignment in the “Assignments” tab.

1. (50 points) XV6 User Program

Part A (30 Points)

- Write a user program, **tail**, that outputs the final ten lines of a file
- If a file name is provided, read from the file.
  - *tail filename*
- Otherwise, read from the standard input.
  - *cat example.txt* | *tail*
- Do not make assumptions on the length or contents of the input file

Examples:

```
$ cat example.txt
  No. 1
  No. 2
  No. 3
  No. 4
  No. 5
  No. 6
  No. 7
  No. 8
  No. 9
  No. 10
  No. 11
  No. 12
$ tail example.txt
  No. 3
  No. 4
  No. 5
  No. 6
  No. 7
  No. 8
  No. 9
  No. 10
  No. 11
  No. 12
cat example.txt | tail
  No. 3
  No. 4
  No. 5
  No. 6
  No. 7
  No. 8
  No. 9
  No. 10
  No. 11
  No. 12
```

Part B (20 Points)

Extend the *tail* user program with more functionality listed below

- -n number: outputs the final number lines of the file.
- -n +number: outputs the rest of the file starting at the number offset from the beginning of the file in lines.
- -c number: outputs the rest of the file starting from the number offset from the end of the file in bytes (outputs the final number characters of the file).
- -c +number: outputs the rest of the file starting from the number offset from the beginning of the file in bytes.
- Note: The origin for counting shall be 1; that is, -c +1 represents the first byte of the file, -c 1 the last. Similarly, -n +1 represents the first line of the file, -n 1 the last.

Examples:

```
$ tail -n 3 example.txt
  No. 10
  No. 11
  No. 12
$ tail -n +8 example.txt
  No. 8
```

```
 No. 9
 No. 10
 No. 11
 No. 12
$ tail -c 3 example.txt  // (Note: There is a space in front of the 12!)
  12
$ tail -c +3 example.txt
 . 1
 No. 2
 No. 3
 No. 4
 No. 5
 No. 6
 No. 7
 No. 8
 No. 9
 No. 10
 No. 11
 No. 12
```

Complete Questions 2, 3, 4, 5, 6, and Academic Honesty Pledge on Anubis