Systems Software

DT211/DT228/DT282 Year 4 - Continuous Assessment 1 (15%)

Due Date: Friday 13th March 2020 @ 23:59

Introduction:

A e-commerce company needs to track changes made to its static html website. Problems have occurred in the past where changes were made to the website incorrectly and it wasn't possible to track who made the changes. The company's CTO has listed the functionality they would like to include in their new business model to offer transparency and accountability for all changes made to the website.

How the website management currently works:
Users have an account on a Debian server, they can login and make changes to the website. All changes will be made under their user accounts. The Debian server is running Apache as the webserver. All changes made to the site /var/www/html appear on the website instantly.

What the CTO wants:

The CTO has offered a list of desired functionality for the new website management model:

1. The company will have an internal Intranet site that is a duplicate copy of the live website. Staff can make changes to the Intranet version of the site and see the changes before it goes live. (This will help prevent content issues and page availability issues for users of the site).
2. The website content should be backed up every night.
3. The changes made to the Intranet version of the site needs to documented. The username of the user, the page they modified and the timestamp should be recorded.
4. The live site needs to be updated based on the changes made to the Intranet site. This should happen during the night. There are a large number of files on the website (5000+), only the files that have changed should be copied to the live site folder.
5. No changes should be allowed to be made to the site while the backup/transfer is happening.
6. If a change needs to be urgently made to the live site, it should be possible to make the changes. (Users shouldn't have write access to the new website folder)

Project Requirements:

a. Create a daemon to continually manage the operation of the requirements listed by the CTO above.
b. Identify new or modified site content and log details of who made the changes, this should be generated as a text file report and stored on the server.
c. Maintain a list of site updates (file)
d. Update the live site every night (After the backup completes).
e. When the backup/transfer begins no user should be able to modify and site content.
f. It must be possible to ask the daemon to backup and transfer at any time.
g. A message queue should be setup to allow all processes to report in on completion of a task. (success or failure)
h. Error logging and reporting should be included (log to file)
i. Create a makefile to manage the creation of the executables.

General Assumptions:

1. The company only has one server
2. Backups can be made to a different folder on the server
3. The Intranet and Live site should be two sub directories of /var/www/html

20%	Project Report (explaining the approach taken and the operation of the application etc.). The project template provided on Brightspace must be used.
75%	C Program Solution (all code and supporting docs uploaded to Brightspace)
5%	**5 minute** video screen recording verbally describing and **demonstrating** the **operation of your solution**. The video must address all the project requirements listed above. The **primary focus** of the video is to **show the operation** of all functionality contained within the solution. The website files provided on Brightspace must be used for the video demo.

Note: The 5 minute video is compulsory and must be included in the submission. Failure to comply with this will result in the grade being withheld until a meeting is arranged with Jonathan to demonstrate the operation of the solution.

Grading Rubric:

	70 +	69 – 60	59 – 50	49 – 40	39 - 0
F1 - System Architecture including makefile (**10%**)	Program follows excellent design principles and demonstrates the proper implementation in the application.	Program good excellent design principles and demonstrates the proper implementation in the application.	Program follows good design principles and demonstrates the proper implementation in the application. Solution contains architectural design flaws and/or demonstrates some principles separate from the application.	Program demonstrates a minimal implementation of design principles. Solution contains architectural design flaws, tight coupling and/or demonstrates principles separate from the application.	Program does not demonstrate the operation of the design principles. Major omissions and lack of understanding of design in the Linux environment.
F2 - Daemon (Setup/ Initialisation/ Management::**5%**)	Daemon setup using an init script implementing the singleton pattern and using header files to store configurable variables. The init script can be used to start/stop the daemon.	Deamon configuration complete with setup / initialization / management included. Some additions needed for a real world server solution.	Daemon setup is operational and working correctly. Some features omitted or not working as expected.	Minimal setup / initialization / management offered to control the operation of the daemon. Some configuration options provided for in the solution.	Minimal or no setup / initialization / management offered to control the operation of the daemon.
F3 - Daemon (Implementation::**10%**)	Background process created and completely decoupled from Terminal and IO. Appropriate error checking and signals used.	Background process created and completely decoupled from Terminal and IO.	Process created. Some issues with the process running in background or with decoupling from Terminal and IO.	Process operates, some of the following not working: not in background, still linked to terminal, issues with file permissions and IO.	Damon process does not operate or does not operate as expected.
F4 - Backup Functionality (**10%**)	Backups architected and implemented correctly and fit for purpose in a commercial environment.	Backups implemented and working correctly. Some additions needed for a real world solution.	Backups working correctly. Some features omitted or not working as expected.	Some aspects of the backups operates, minimal attempt.	No backups implemented correctly or not fit for purpose.
F5 - Transfer Functionality (**10%**)	Transfers architected and implemented correctly and fit for purpose in a commercial environment.	Transfers implemented and working correctly. Some additions needed for a real world solution.	Transfers working correctly. Some features omitted or not working as expected.	Some aspects of the transfers operates, minimal attempt.	No transfers implemented correctly or not fit for purpose.
F6 - Lockdown folder for Backup / Transfer (**5%**)	The folders are completely locked down during the backup / transfer process and reverts to standard permissions when the process is complete.	The folders are completely locked down during the backup / transfer process and reverts to standard permissions when the process is complete. Some minor issues with the implementation or operation of the solution.	The folders are completely locked down during the backup / transfer process and reverts to standard permissions when the process is complete. Some features omitted or not working as expected.	The implementation of the lockdown is provided but not operating as expected. Major omissions and/or not operating as expected.	Minimal implementation and not fit for purpose.

F7 - Reporting (IPC::**10%**)	Reporting implemented correctly for all aspects of the program.	Reporting implemented correctly for all most of the program.	Reporting implemented correctly for some aspects of the program.	Minimal Reporting implemented in the program.	No Reporting implemented correctly for all aspects of the program or not fit for purpose.
F8 – Logging and Error Logging (**15%**)	Error logging and implemented correctly for all aspects of the program.	Error logging implemented correctly for all most of the program.	Error logging implemented correctly for some aspects of the program.	Minimal Error logging implemented in the program.	No error logging implemented correctly for all aspects of the program or not fit for purpose.
F9 - Documentation (**20%**)	The documentation is well written and clearly explains all architectural choices and functionality of the system	The documentation is well written. Could have explained the code and the principle in more detail.	The documentation is acceptable. Could have explained the code and the principle in more detail. Omissions of content or misinterpretation of the principle demonstrated.	The documentation is minimal or not focused on the problem description. Could have explained the code and the principle in more detail. Omissions of content or misinterpretation of the principles demonstrated.	The documentation is simply comments embedded in the code and does explain the code or the principle. Minimal attempt in all aspects.
F10 - Exemplar Video (**5%**)	Video is well prepared and shows and describes the exact operation of the solution. Complex aspects of the solution have been described in good detail and within the 5-minute maximum.	Video is well prepared and shows and describes the exact operation of the solution and within the 5-minute maximum.	The video is acceptable. Could have explained the code and the solution in more detail. Minor omissions of content or detail in the video demonstration and within the 5-minute maximum.	The video is minimal or not focused on the problem description. Could have explained the code and the solution offered in more detail. Omissions of content or detail in the video demonstration.	Video doesn't capture the operation of the solution and/or doesn't offer a verbal description of the functionality of the system from a code perspective.

Notes:

- All of the above features (F1 to F8) are being evaluated based on their ability to implement the feature and the quality of the code solution provided.
- The system should be configured to run when the system boots. (/etc/init.d/skeleton)(/usr/sbin)
- The above grading scheme is describing features and functionality fully incorporated into the daemon solution.
- The Exemplar Video must follow the structure detailed below.
- The Exemplar Video is compulsory. If it is not possible to create a video a meeting will need to be arranged with Jonathan to demonstrate the operation of the application. Failure to comply with this will result in a zero grade being applied to this submission.
- Penalties will be applied to late submissions (see Brightspace for details on late penalties)

Video Structure:

- Demonstrate your build automation using make.
- Use the following bootstrap template as your website: https://startbootstrap.com/themes/creative/
- The video should **demonstrate the operation of the features (F1 to F8)** listed in the grading rubric above.
- In the video each of the features should be named and its functionality demonstrated as a **screen recording**.
- The **main focus** of the video is to demonstrate the **operation of the application**, the document can be used to describe choices made in the implementation etc.
- The **duration** of the video **should not exceed 5 mins**, this should be strictly adhered to.