

Systems Software

TU856/TU857/TU858 Year 4 - Assignment 1 (15%)

Deadline: Monday 10th March 2025 10:00am

Introduction:

A manufacturing company has been experiencing difficulties getting current information from specific departments to see the current state of the plant. At certain points in the past manufacturing had to stop due to lack of raw materials and limited space in the loading bay. The managers from the following departments need to complete a report on a daily basis to feed into a dashboard reporting system:

- Warehouse
- Manufacturing
- Sales
- Distribution

The department managers have to upload the xml report to a shared directory for collection by the dashboard system. Problems have occurred in the past where changes were made to the shared directory 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 shared directory.

How the website management currently works:

The above departments are responsible for reporting on a daily basis. Reports need to be submitted for inclusion in the daily dashboard report. Linux Managers have an account on a Linux server, they can login and make changes to the shared directory. All changes will be made under their user accounts. There have been disputes in the past over who is responsible for filing reports and for reports allegedly disappearing from the shared directory.

What the CTO wants:

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

1. The company will offer a new shared directory for the 4 departments to upload their xml reports. Authorised users can make changes to directory. The reports will be collected on a nightly basis at 1am and moved to the directory for the dashboard system for processing.
2. The dashboard directory content should be backed up every night.
3. The changes made to the department managers upload directory needs to be documented. The username of the user, the file they modified and the timestamp should be recorded.
4. No changes should be allowed to be made to the directories (upload and reporting) while the backup/transfer is happening.
5. If a change needs to be urgently made to the live site, it should be possible to make the changes. (No users shouldn't have write access to the new reporting directory)

Project Requirements:

- a. Create a daemon to continually manage the operation of the requirements listed by the CTO above.
- b. Identify new or modified xml reports and log details of who made the changes, this should be generated as a text file report and stored on the server.
- c. The department managers must upload their xml report file by 11.30pm each night. The uploads to the shared directory must be moved to the reporting directory. This should happen at 1am. (This functionality needs to be self-contained in the solution. Do NOT use cron)
- d. If a file wasn't uploaded this should be logged in the system. (A naming convention can be used to help with this task.)
- e. When the backup/transfer begins no user should be able to modify either the upload or reporting directory.
- f. It must be possible to ask the daemon to backup and transfer at any time.
- g. IPC 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 for all main tasks in the solution.
- 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 directory on the server
3. The upload and reporting directories should be two sub directories. (You can choose this location in your system)

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%	<u>5 minute</u> video screen recording verbally describing and <u>demonstrating</u> the <u>operation of your solution</u> . The video must address all the project requirements listed above. The <u>primary focus</u> of the video is to <u>show the operation</u> 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 to demonstrate the operation of the SUBMITTED solution in person.

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 follows 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 directories for Backup / Transfer (5%)	The directories are completely locked down during the backup / transfer process and reverts to standard permissions when the process is complete.	The directories 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	The directories 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 - Process management and IPC (IPC::10%)	Very good usage of the exec() family of functions replaces the current process image with a new process image.	operation of the solution. Good usage of the system()/exec() family of functions replaces the current process image with a new process image.	Good usage of the system()/exec() family of functions replaces the current process image with a new process image. Some features omitted or not working as expected.	Minimal use of forking to facilitate additional functionality running is a different process.	Minimal implementation and 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 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.

Video Structure:

- Demonstrate your build automation using make.
- 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.