CYBR371

Assignment 1

Olivia Fletcher
300534281
fletcholiv

TASK 1

Staff Access Matrix [10 Marks]

Represent the policy using an access matrix. Associated rights are based on UNIX access rights and include: Read, Write and Execute. This must include identification of all subjects/objects and appropriate privileges/permissions.

FILES	sbasicinfo.log	pbasicinfo.log	pmedicalrecord.log
Administrators	Read, Write and Execute [rwx]	Read, Write and Execute [rwx]	Read, Write and Execute [rwx]
Registered Doctor	Read [r–]	Read [r–]	Read, Write [rw-]
Non-Registered Doctor	Read [r–]	Read [r–]	No permissions [---]
Nurses	Read [r–]	Read [r–]	Read [r–]
Receptionists	Read [r–]	Read and Write [rw-]	No permissions [---]

SCRIPTS	file-system-setup.sh	staff-create.sh	acls.sh
Administrators	Read, Write and Execute [rwx]	Read, Write and Execute [rwx]	Read, Write and Execute [rwx]
Registered Doctor	No permissions [---]	No permissions [---]	No permissions [---]
Non-Registered Doctor	No permissions [---]	No permissions [---]	No permissions [---]
Nurses	No permissions [---]	No permissions [---]	No permissions [---]
Receptionists	No permissions [---]	No permissions [---]	No permissions [---]

SCRIPTS	register-patient.sh	search-patient.sh	search-doctor.sh
Administrators	Read, Write and Execute [rwx]	Read, Write and Execute [rwx]	Read, Write and Execute [rwx]
Doctors	No permissions [---]	No permissions [---]	No permissions [---]
Non-Registered Doctor	No permissions [---]	No permissions [---]	No permissions [---]
Nurses	No permissions [---]	No permissions [---]	No permissions [---]
Receptionists	Read and Execute [r-x]	Read and Execute [r-x]	Read and Execute [r-x]

SCRIPTS	visit.sh	check-medication.sh	audit.sh
Administrators	Read, Write and Execute [rwx]	Read, Write and Execute [rwx]	Read, Write and Execute [rwx]
Doctors	Read and Execute [r-x]	Read [r–]	No permissions [---]
Non-Registered Doctor	No permissions [---]	No permissions [---]	No permissions [---]
Nurses	No permissions [---]	Read and Execute [r-x]	No permissions [---]
Receptionists	No permissions [---]	No permissions [---]	No permissions [---]

TASK 2

File System Setup Script [5 Marks]

Write a Bash/Python script (file-system-setup.sh) to create the file system directory structure according to the information given in the case study. Only the administrator can execute this script. This script must follow the principle of least privilege.

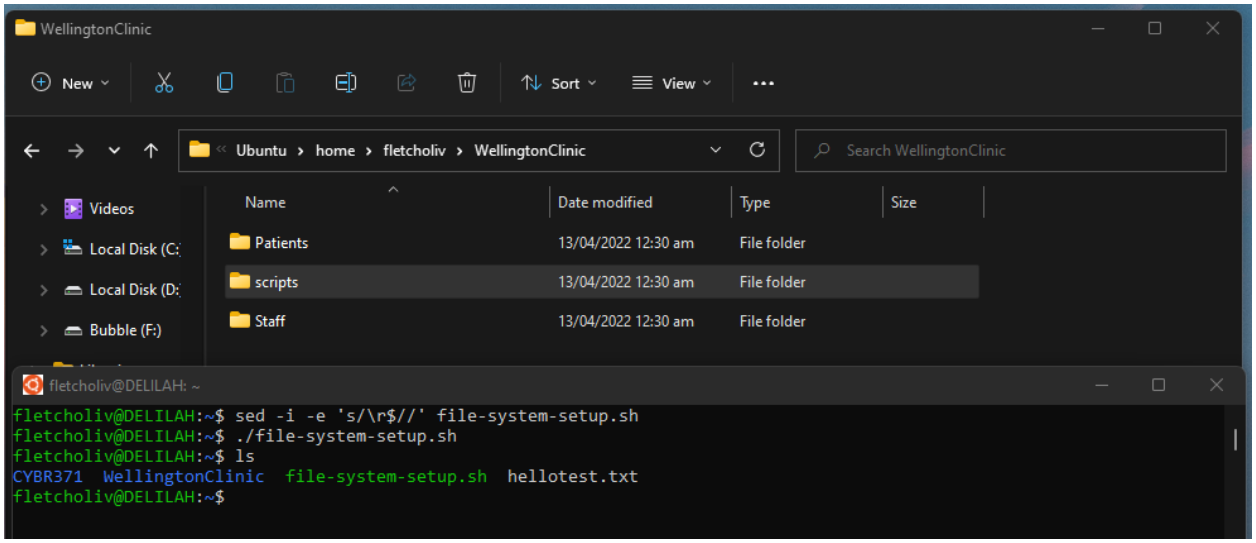
file-system-setup.sh - Notepad

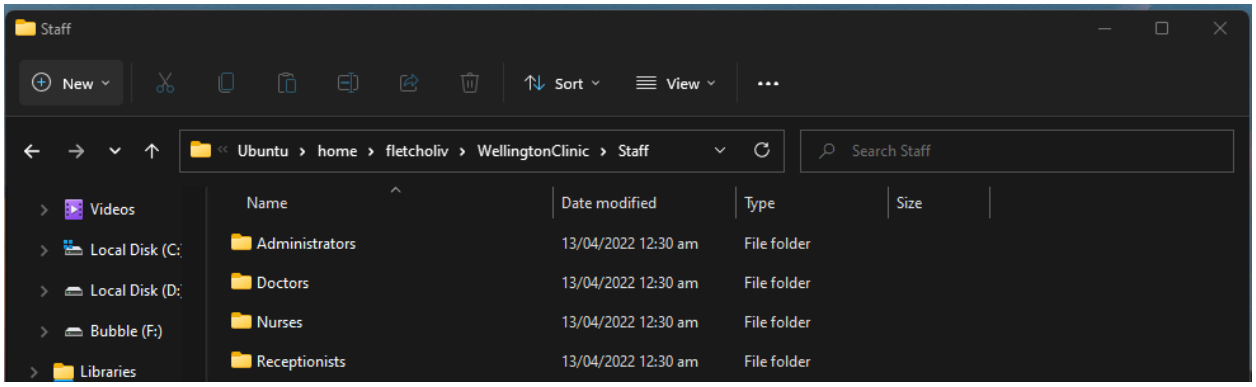
File Edit View

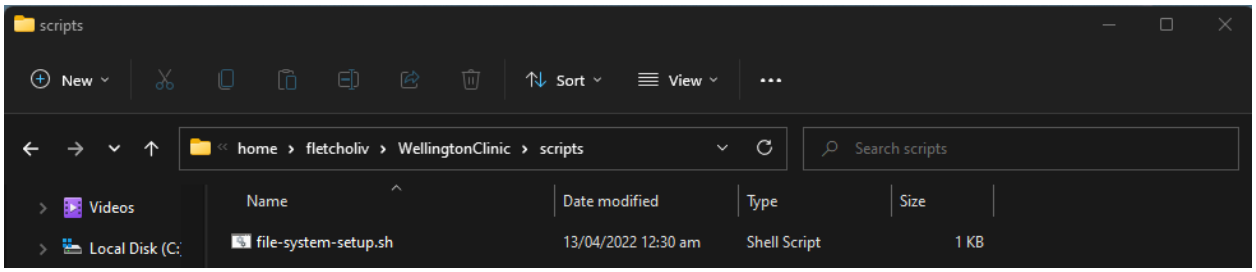
```
#!/bin/bash

# Task  2 - [5 marks]

# Write a script [file-system-setup.sh] which creates a file system directory structure
# Only an admin can execute this script

# Make the clinic directory and enter it
mkdir WellingtonClinic
      cd WellingtonClinic

      # Make the necessary directories for the clinic
      mkdir Staff
      mkdir Patients
      mkdir scripts

            # Enter the staff directories and create the necessary directories for staff
            cd Staff
            mkdir Administrators
            mkdir Doctors
            mkdir Nurses
            mkdir Receptionists
            cd

      # Copy the script file into the clinics scripts folder
      cp file-system-setup.sh WellingtonClinic/scripts
```

Ln 4, Col 1 100% Windows (CRLF) UTF-8

- After running the above script the directories are successfully created and the script file has successfully copied into the scripts directory



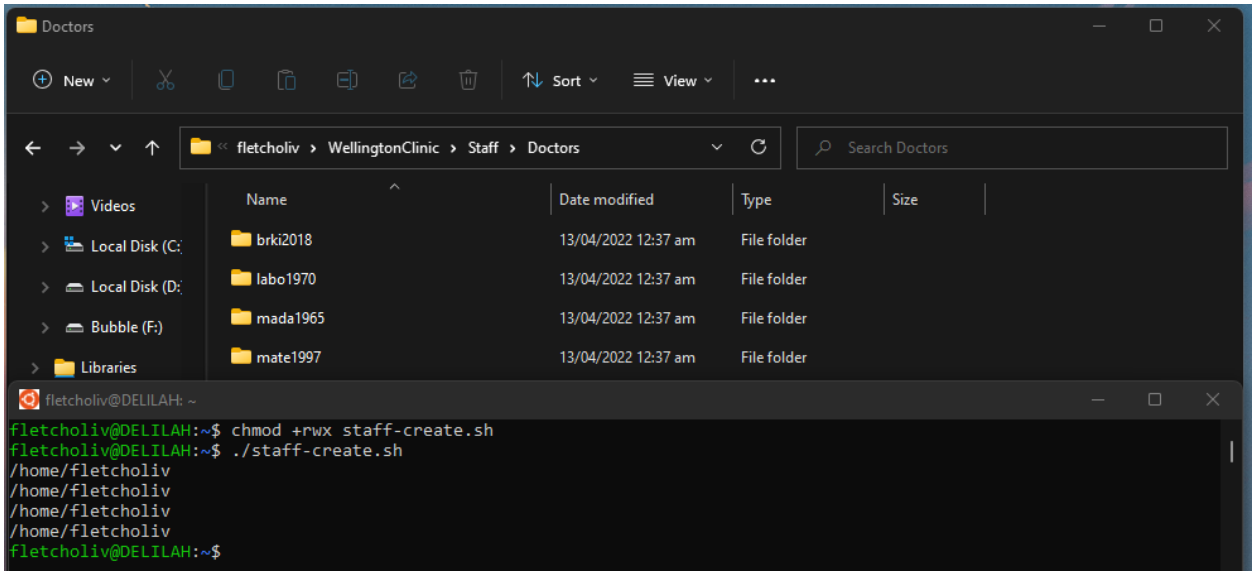




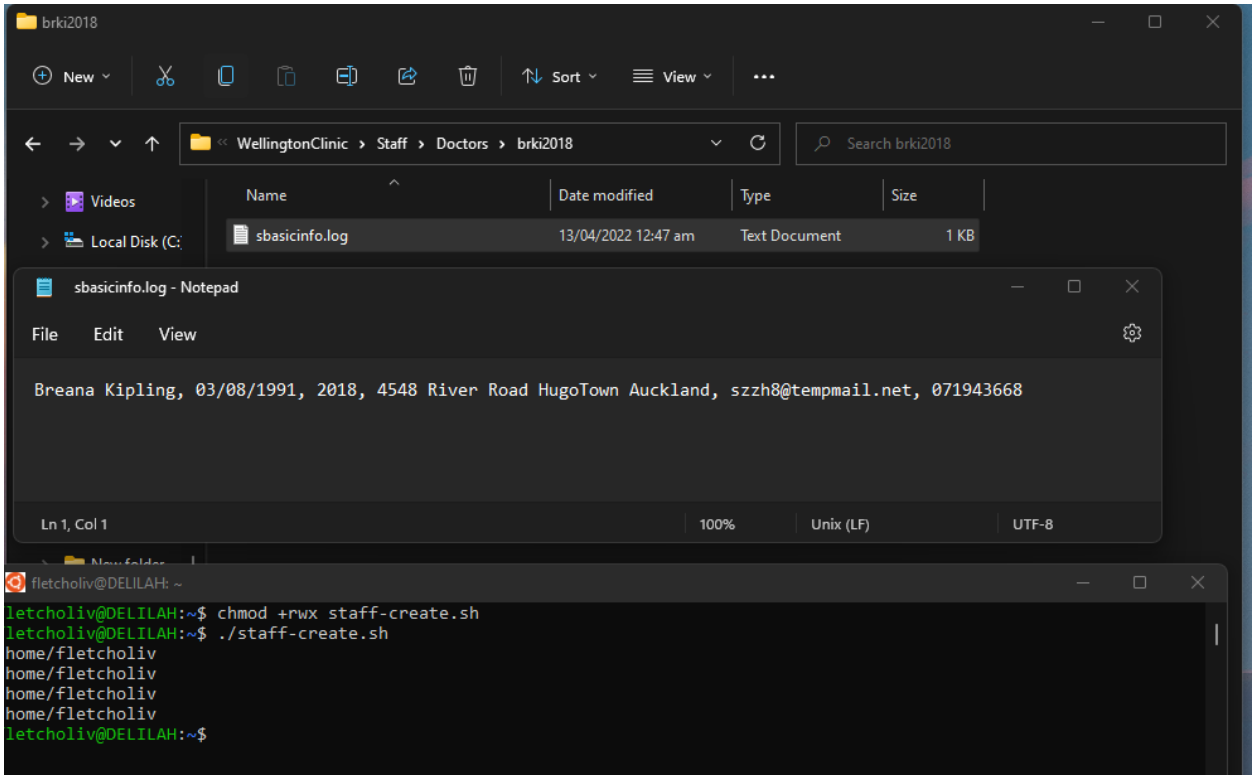
TASK 3

Create Staff Script [5 Marks]

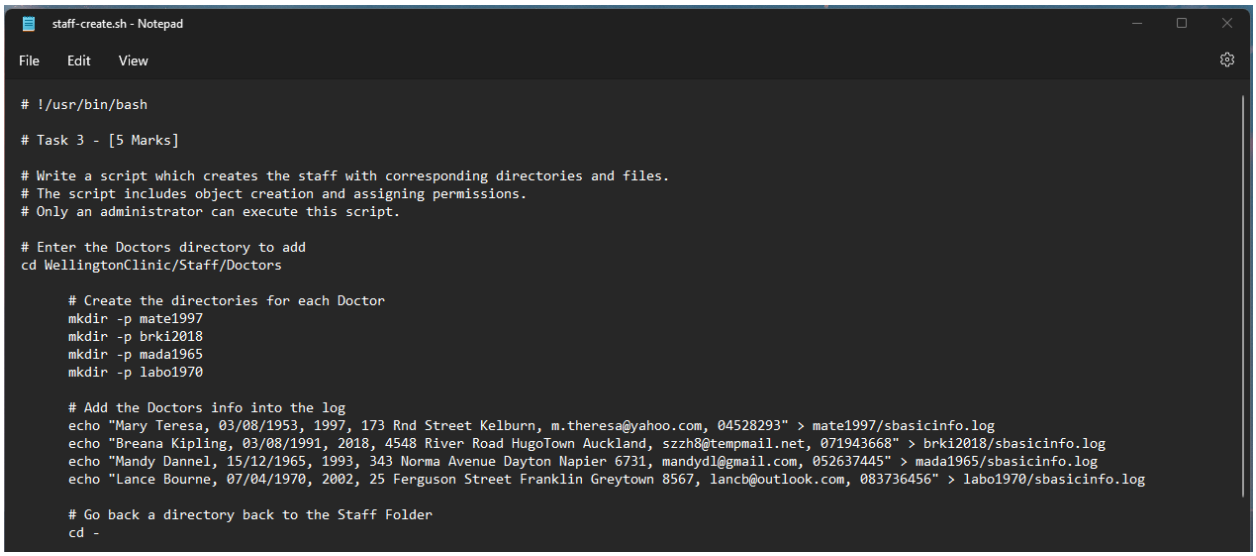
Write a Bash/Python script (staff-create.sh) to create the following staff and corresponding directories and files associated with each. The script should illustrate (include) object creation, assigning permissions. Only administrators are able to execute this script. This information must be embedded in the script and must not require user input.



- Above showing the running of the script and the staff directories successfully were created, below opening a random staff .log file to show the data was successfully created and stored in the staff members respective sbasicinfo.log file



- For each staff member I created their own folder using their username in their respective directory (doctor/nurse etc) and added their data to the sbasicinfo.log file in their own folder
- Doctors;



- Nurses;

```
cd WellingtonClinic/Staff/Nurses

      # Create the directories for each Nurse
      mkdir -p lubl1980
      mkdir -p phmc2008

      # Add the Nurses info into the log
      echo "Lucia Blakeley, 1/09/1980, 2004, 935 Massachusetts Avenue Hamilton 4562, lucyblak@outlook.com, 38347463" > lubl1980/sbasicinfo.log
      echo "Phil McGraw, 5/04/1984, 2008, 5 Razyn Street Petone Lower Hutt 8435, philmcg@gmail.com, 02384756" > phmc2008/sbasicinfo.log

      # Go back a directory back to the Staff Folder
      cd -
```

- Receptionists;

```
cd WellingtonClinic/Staff/Receptionists

      # Create the directories for each Receptionist
      mkdir -p ansm1974
      mkdir -p lubl1980
      mkdir -p phmc2008

      # Add the Receptionists info into the log
      echo "Andy Smith, 13/09/1974, 2002, 58 Foster avenue Wellington 5011, smith.andy74@gmail.com, 07284756" > ansm1974/sbasicinfo.log
      echo "Lucia Blakeley, 1/09/1980, 2004, 935 Massachusetts Avenue Hamilton 4562, lucyblak@outlook.com, 38347463" > lubl1980/sbasicinfo.log
      echo "Phil McGraw, 5/04/1984, 2008, 5 Razyn Street Petone Lower Hutt 8435, philmcg@gmail.com, 02384756" > phmc2008/sbasicinfo.log

      # Go back a directory back to the Staff Folder
      cd -
```

- Administrators;

```
cd WellingtonClinic/Staff/Administrators

      # Create the directory for the Administrator
      mkdir -p pasa1993

      # Add the Administrator info into the log
      echo "Pauline Sanderson, 08/03/1993, 1995, 2452 Randolph Street Bedford Auckland 7752, paulsand@admins.co.nz, 03747543" > pasa1993/sbasicinfo.log

      # Go back a directory back to the Staff Folder
      cd -

# Copy the script file into the clinics scripts folder
cp file-system-setup.sh WellingtonClinic/scripts
```

- Also constructed groups for each staff member, creating group for associated department and created and added the staff users to their associated groups.

```
      # Creating group for Doctors
      sudo groupadd -f doctors

      # Creating each Doctor user
      sudo useradd mate1997
      sudo useradd brki2018
      sudo useradd mada1965
      sudo useradd labo1970

      # Adding each Doctor user to group
      usermod -G doctors mate1997
      usermod -G doctors brki2018
      usermod -G doctors mada1965
      usermod -G doctors labo1970
```

```
      # Creating group for Nurses
      sudo groupadd -f nurses

      # Creating each Nurse user
      sudo useradd lubl1980
      sudo useradd phmc2008

      # Adding each Nurse user to group
      usermod -G nurses lubl1980
      usermod -G nurses phmc2008
```

```
# Creating group for Receptionists
sudo groupadd -f nurses

# Creating each Receptionist user
sudo useradd ansm1974
sudo useradd lubl1980
sudo useradd phmc2008

# Adding each Receptionist user to group
usermod -G nurses ansm1974
usermod -G nurses lubl1980
usermod -G nurses phmc2008
```

```
# Creating group for the Administrator
sudo groupadd -f administrators

# Creating the Administrator user
sudo useradd pasa1993

# Adding the Administrator user to group
usermod -G administrators pasa1993
```

TASK 4

ACL Information [5 Marks]

Explain in detail where the ACL information of an object is saved on a linux system (EXT2, 3 and 4 file systems) and how your system keeps track of them.

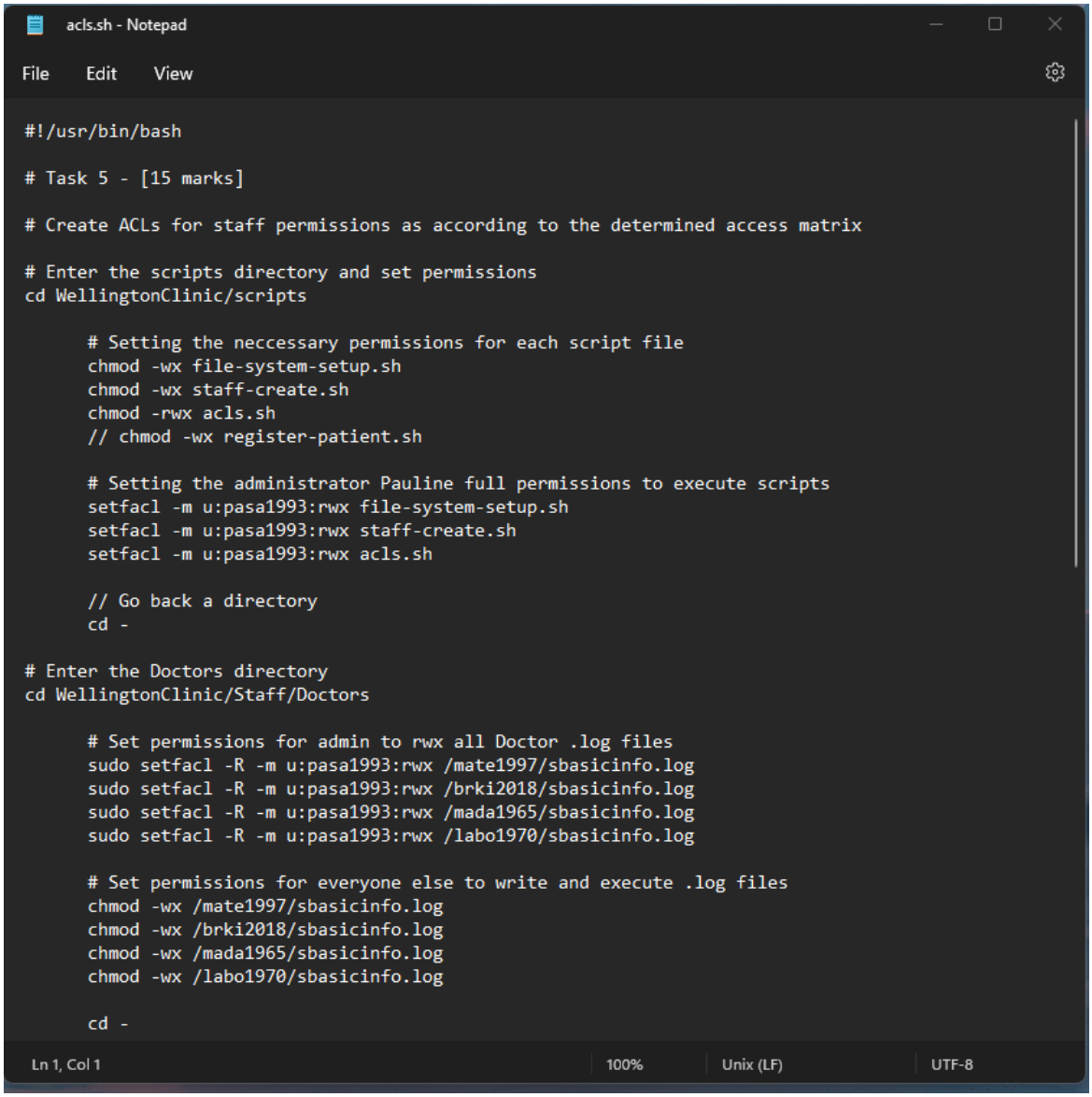
- Similar to filename metadata the ACL information is not necessarily stored as a part of the file. They are stored as a separate data block on the disk and our system keeps track of it in the file inodes using the EA's (extended attributes) mechanism; the ACLs information are implemented as the EA's on linux.

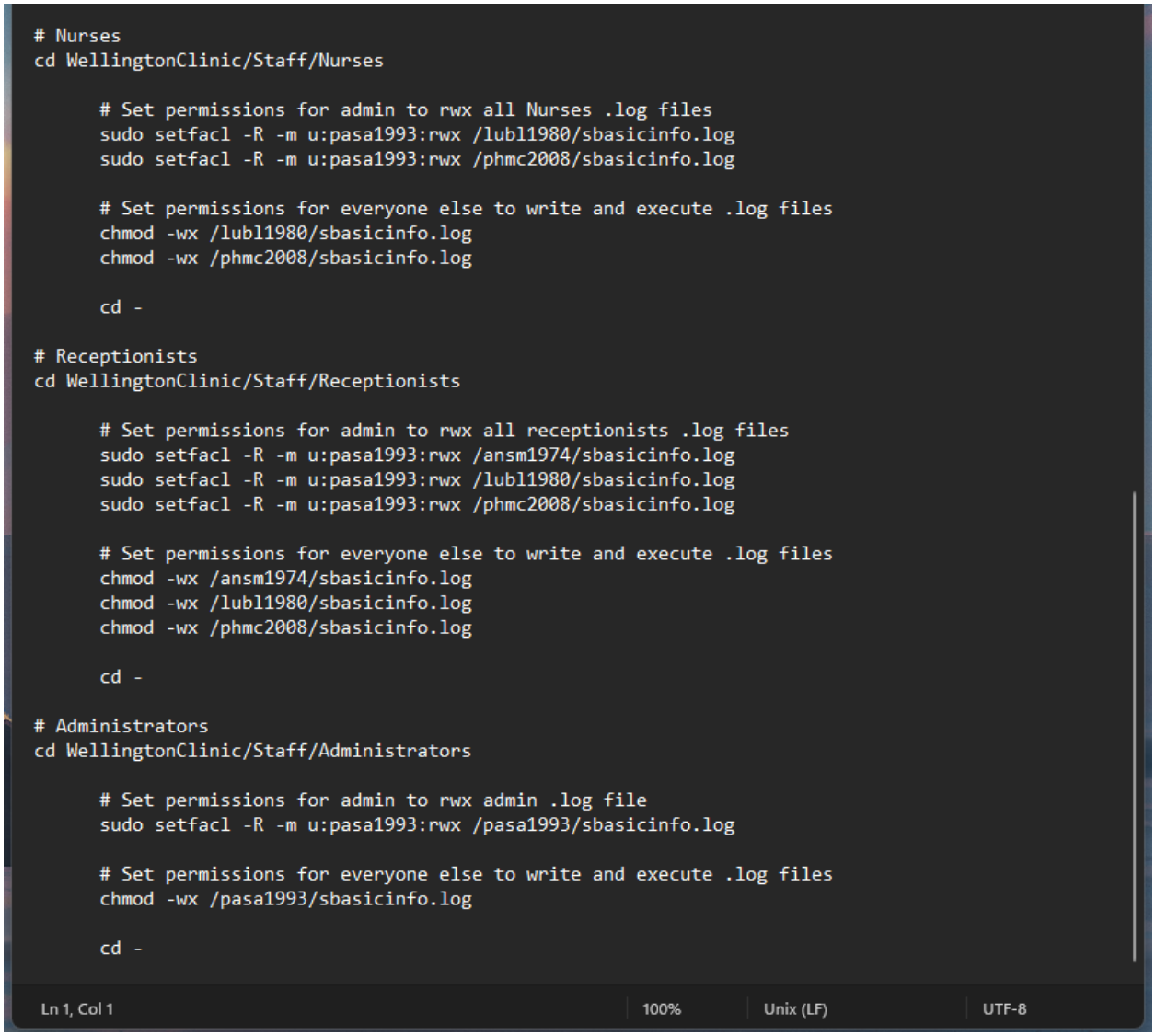
TASK 5

Create ACL Script [15 Marks]

Write a Bash/Python script (acls.sh) to assign rights using ACLs according to the policy rules and the provided and determined access matrixes (File and directory access matrixes). The script must illustrate (include) create of ACLs according to the use case and must follow the principle of least privilege.

- Using the access matrix as reference in the scripts directory I set all the scripts to their necessary permissions, write and execute for file-system-setup and staff-create and read, write and execute for the acls.
- Set pauline admin to full permissions for each of the scripts
- In each of the staffs directories (doctors,nurses etc) I set pauline admin full permissions to each members sbasicinfo.log file and set basic write execute for everyone else


acls.sh - Notepad
File
Edit
View
#!/usr/bin/bash
Task 5 - [15 marks]
Create ACLs for staff permissions as according to the determined access matrix
Enter the scripts directory and set permissions
cd WellingtonClinic/scripts
Setting the neccessary permissions for each script file
chmod -wx file-system-setup.sh
chmod -wx staff-create.sh
chmod -rwx acls.sh
// chmod -wx register-patient.sh
Setting the administrator Pauline full permissions to execute scripts
setfacl -m u:pasa1993:rwx file-system-setup.sh
setfacl -m u:pasa1993:rwx staff-create.sh
setfacl -m u:pasa1993:rwx acls.sh
// Go back a directory
cd -
Enter the Doctors directory
cd WellingtonClinic/Staff/Doctors
# Set permissions for admin to rwx all Doctor .log files
sudo setfacl -R -m u:pasa1993:rwx /mate1997/sbasicinfo.log
sudo setfacl -R -m u:pasa1993:rwx /brki2018/sbasicinfo.log
sudo setfacl -R -m u:pasa1993:rwx /mada1965/sbasicinfo.log
sudo setfacl -R -m u:pasa1993:rwx /labo1970/sbasicinfo.log
# Set permissions for everyone else to write and execute .log files
chmod -wx /mate1997/sbasicinfo.log
chmod -wx /brki2018/sbasicinfo.log
chmod -wx /mada1965/sbasicinfo.log
chmod -wx /labo1970/sbasicinfo.log
cd -
Ln 1, Col 1
100%
Unix (LF)
UTF-8



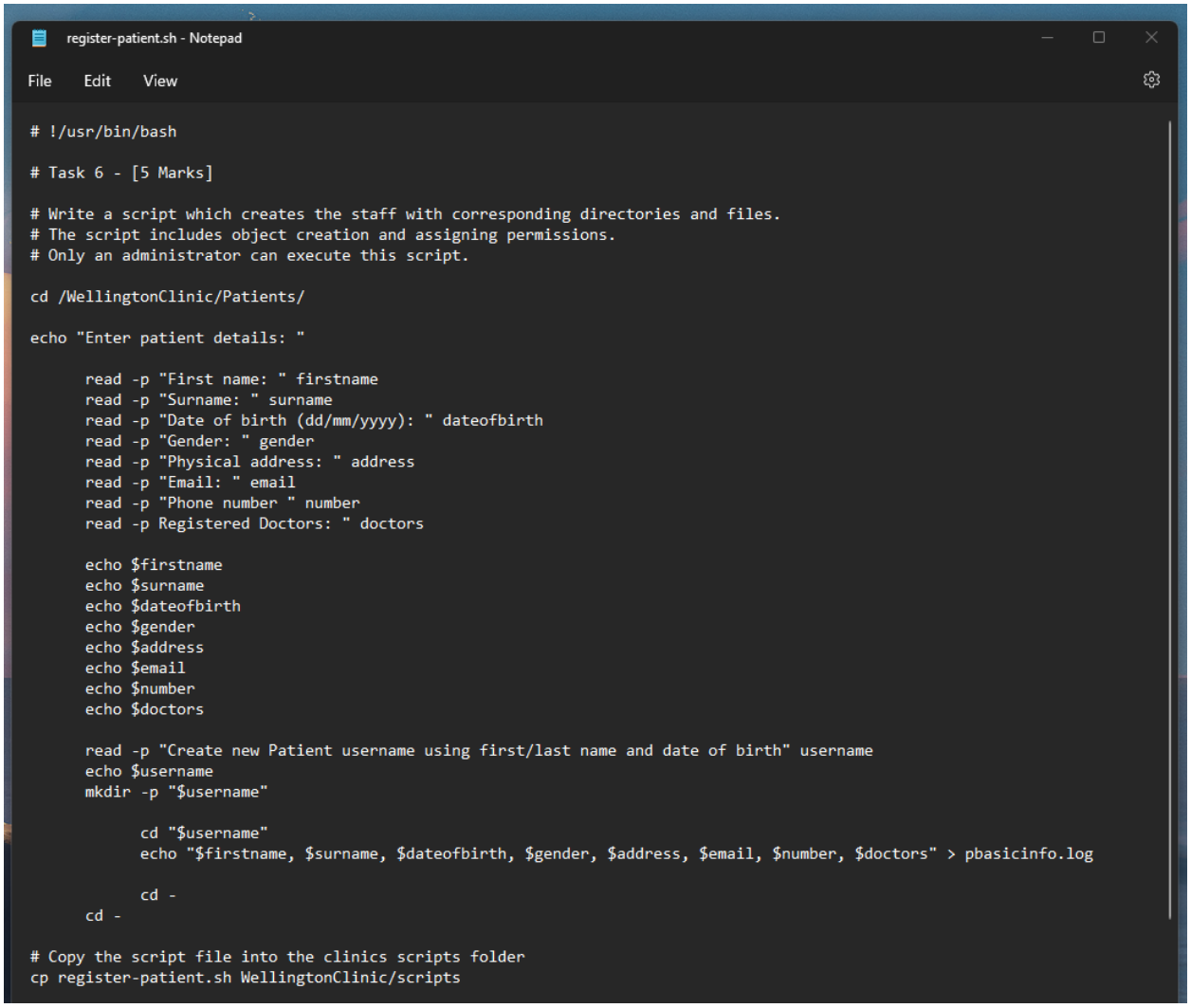
```
# Nurses
cd WellingtonClinic/Staff/Nurses

      # Set permissions for admin to rwx all Nurses .log files
      sudo setfacl -R -m u:pasa1993:rwx /lubl1980/sbasicinfo.log
      sudo setfacl -R -m u:pasa1993:rwx /phmc2008/sbasicinfo.log

      # Set permissions for everyone else to write and execute .log files
      chmod -wx /lubl1980/sbasicinfo.log
      chmod -wx /phmc2008/sbasicinfo.log

      cd -

# Receptionists
cd WellingtonClinic/Staff/Receptionists

      # Set permissions for admin to rwx all receptionists .log files
      sudo setfacl -R -m u:pasa1993:rwx /ansm1974/sbasicinfo.log
      sudo setfacl -R -m u:pasa1993:rwx /lubl1980/sbasicinfo.log
      sudo setfacl -R -m u:pasa1993:rwx /phmc2008/sbasicinfo.log

      # Set permissions for everyone else to write and execute .log files
      chmod -wx /ansm1974/sbasicinfo.log
      chmod -wx /lubl1980/sbasicinfo.log
      chmod -wx /phmc2008/sbasicinfo.log

      cd -

# Administrators
cd WellingtonClinic/Staff/Administrators

      # Set permissions for admin to rwx admin .log file
      sudo setfacl -R -m u:pasa1993:rwx /pasa1993/sbasicinfo.log

      # Set permissions for everyone else to write and execute .log files
      chmod -wx /pasa1993/sbasicinfo.log

      cd -
```


TASK 6

Create Patient Registry Script [5 Marks]

Write a Bash/Python script (resgister-patient.sh) to register a new patient by creating the necessary folders and files. Only the receptionist (including the designated nurse) must be able to execute the script. The script must not allow a doctor to assign themselves to be the registered doctor. The script asks the receptionist to enter all the patient's basic personal information.

- Using the echo and read function I prompt the user (administrator in this case) to add the patient details, set their username and add their information ot their associated pbasicinfo.log file

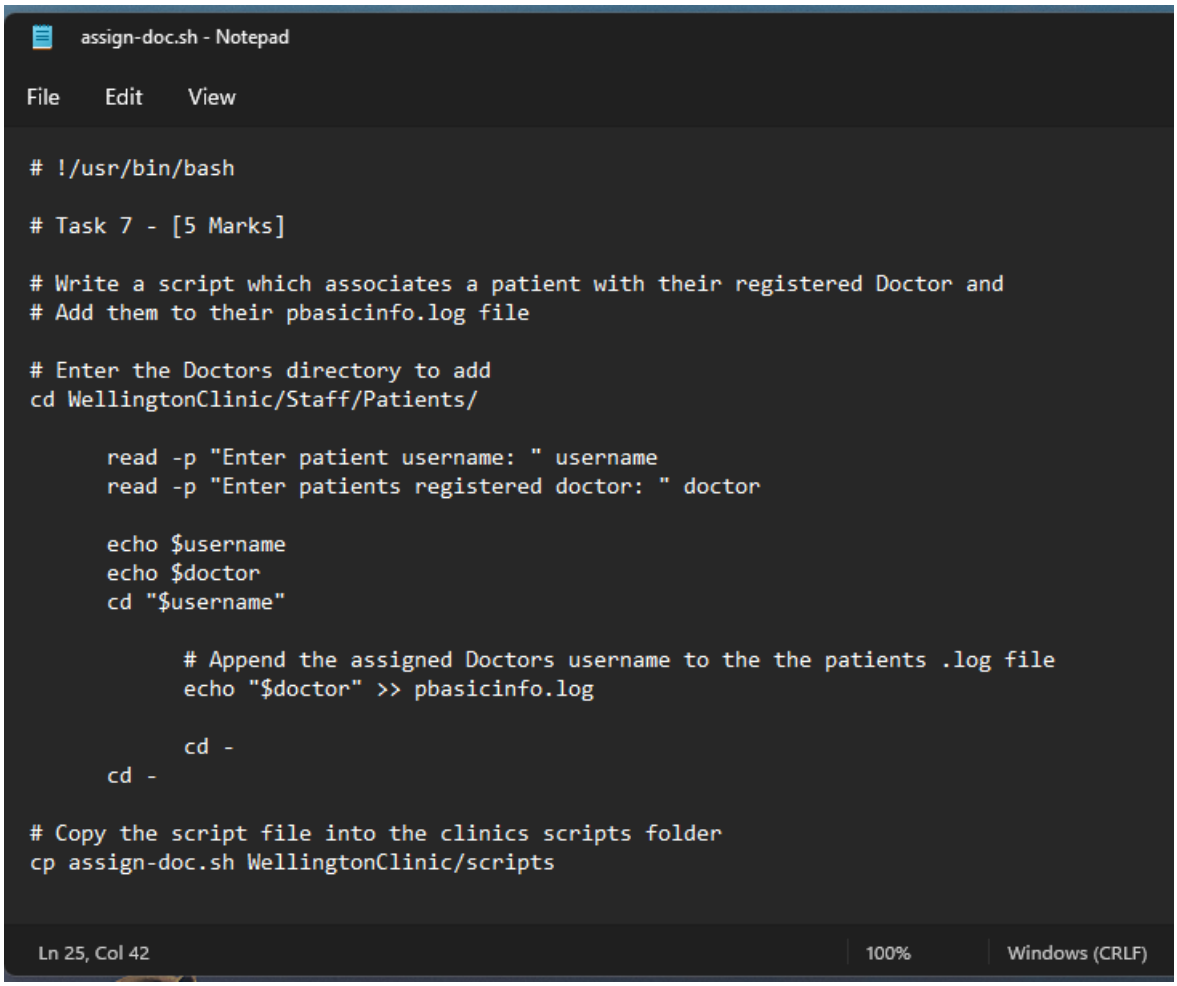


TASK 7

Create Doctor Assigning Script [5 Marks]

Write a script (assign-doc.sh) to add a doctor to the list of registered doctors for a patient. Only the receptionist (including the designated nurse) can execute this script. The information is written to pbasicinfo.log file. The script must ask the user to enter the name of the patient and the name or username of the newly registered/assigned doctor.

- I created a simple script which simply asks the user (receptionist in this case) to enter the patient and doctors username
- The doctors username is then added to the patients associated pbasicinfo.log file



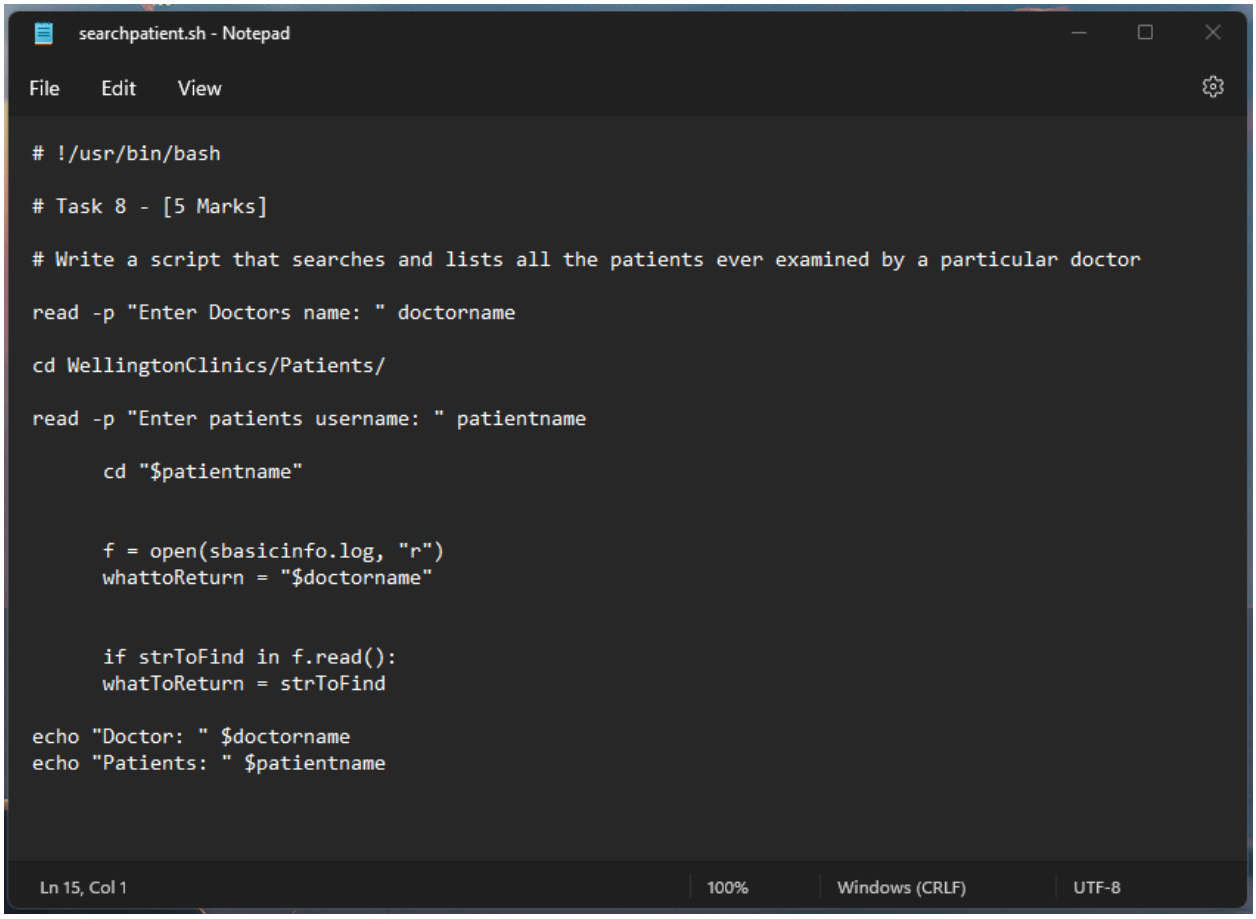
TASK 8

Create Patient Search Script [5 Marks]

Write a script (searchpatient.sh) to search and list all the patients ever examined by a particular doctor. This script can be executed by the receptionist only (excluding those who are also nurses). The script asks for the full name or username of a particular doctor. The script must display information as follows:

Doctor	Patients
Mandy Dannel (mada1993)	David Travert, Lance Bourne

- Reference taken from;
- https://www.codegrepper.com/code-examples/python/check+if+file+contains+string+python
- Attempted // Ask for Doctors username and Patients username, enter the patients directory and check the patients log file to see if it contains the doctors username



TASK 9

Create Doctor Search Script [5 Marks]

Write a script (searchdoctor.sh) to search and list all the doctors who have ever been assigned to a patient. This script can be executed by the receptionist only (excluding those who are also nurses). The script must display the information as follows:

Patient	Doctors
Lance, Bourne	mada1993, Mary Teressa:mate1997,Breana Kipling:brk2018

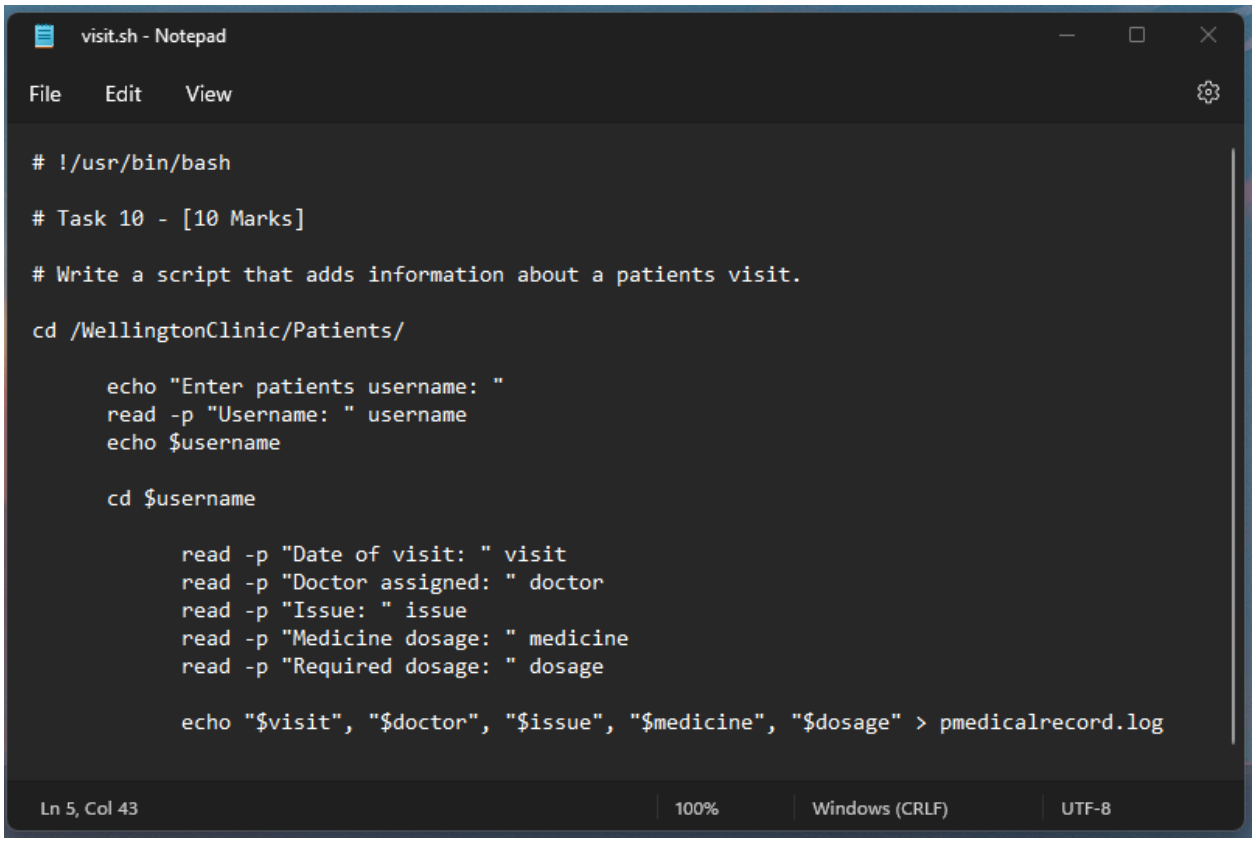
- Similar to above script, task 8?

TASK 10

Create Patient Visit Script [10 Marks]

Write a Bash/Python script (visit.sh) to add information about a patient's visit. When a doctor executes this script and the doctor is in the list of registered doctors, it allows the user to add an entry to the records (please see pmedicalrecord.log for the information and the format). Therefore this script must only allow a doctor to write to a patients' pmedicalrecord.log file if the doctor executing it is in the list of the registered doctors for the patient.

- Simple script similar to register patient where I ask the user (in this case the Doctor) to enter the patients username, we enter the patients directory and then create a new .log file that contains the patient's visit information

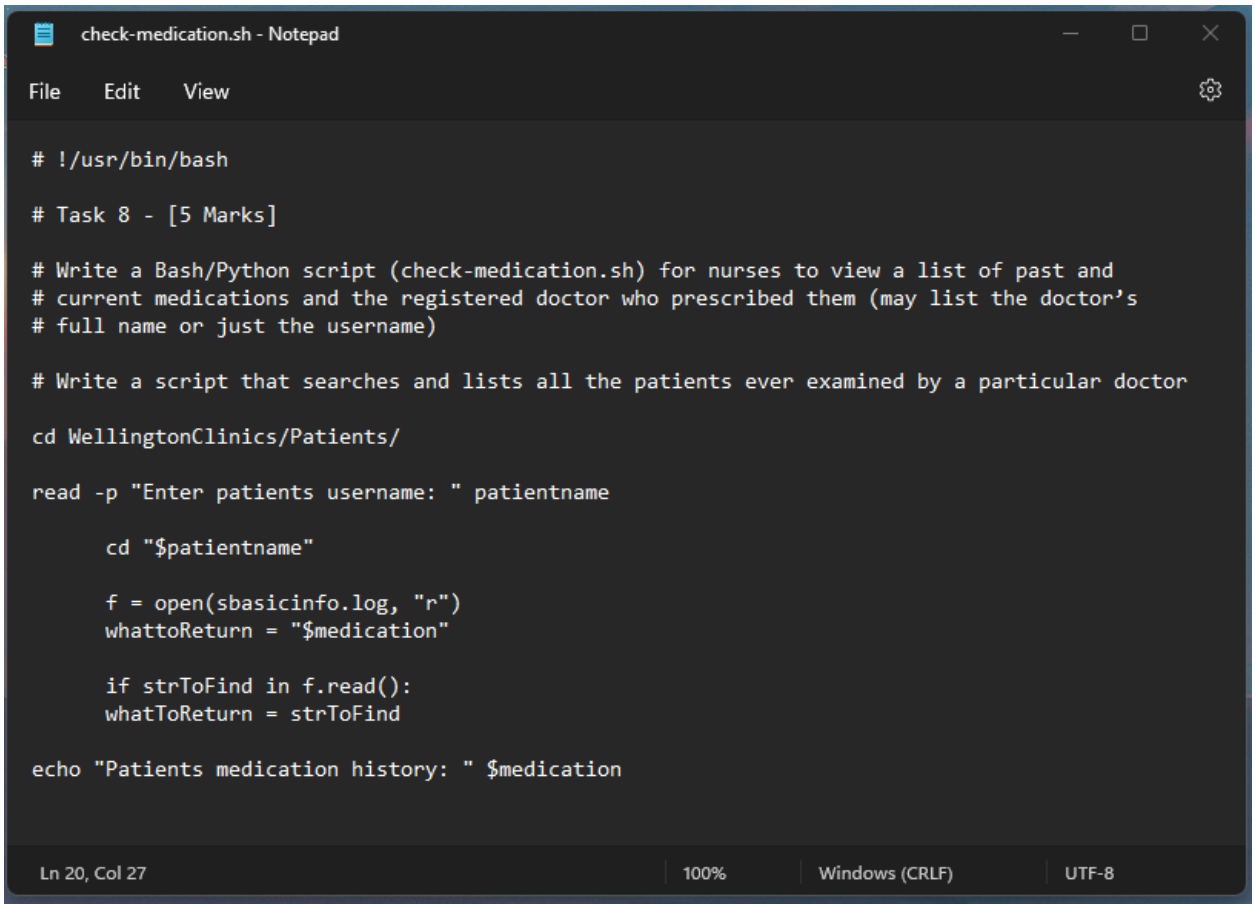


TASK 11

Create Medication Check Script [5 Marks]

Write a Bash/Python script (check-medication.sh) for nurses to view a list of past and current medications and the registered doctor who prescribed them (may list the doctor's full name or just the username)

- Similar to the search patient and search doctor scripts I am checking and printing the medications and registered doctors history on a specified patient .log file



TASK 12

Create Audit Script [10 Marks]

A script file (audit.sh) which displays information on any changes in permission, object creation, deletion and modification on the WellingtonClinic directory, its subdirectories and files. The audit script should display the username performing the operation, the type of operation, the object on which the operation takes place and the data associated with the access operation. This script is only run by the administrator(s). (See below).

Username	Object	Operation	Date
Mate1997	pmedicalrecord.log	write	date

- // Not attempted

TASK 13

Alternative Methods to Minimize Potential Confidentiality Violation [10 Marks]

Nurses having r (read) access to the pmedicalrecord.log file is a serious vulnerability even though access to information is filtered through the interface (i.e check-mdeication.sh) to view the permitted information within the file only. Propose alternative methods to minimize the risk of confidentiality violation of patient's medical history information and justify the proposed method(s). Discuss in detail how your method would achieve that, implement your solution, and illustrate its effectiveness.

- // Not attempted script but;
- If I were to try this I would go down the route similar to the registered doctor for patients and only the nurses that are currently needing access to the patients information can view.
- This function will then reset after each time so once the nurse is finished with the patient's information they are not able to go back and read again so permissions go from [r–] to [---].

TASK 14

Comments [5 Marks]

Writing, presentation, and Code readability (i.e. structured code, inclusion of #comments)