



ENCS3130
Project report

Rana Musa
1210007
Section 4

Brief explanation about code

```
import subprocess
from xml.dom import minidom
import os

class CommandManualGenerator(): #to read commands from the file
    def __init__(self,file):
        self.file=file
    def generate_manual(self):
        self.file = open(f"{self.file}", "r")
        self.content = self.file.read()
        self.command_list = self.content.split("\n")
        return self.command_list
```

At first, I imported some libraries such as subprocess, xml. I added a class to CommandManualGenerator to read the input file that has a command list and put them in a list to become easier to handle.

```
17 class CommandManual(): #to generate each command manual
18     def __init__(self, command):
19         self.command = command
20
21     def generate_description(self, command): #getting the description
22         des = f"man {command} | awk '/^DESCRIPTION/,/^$/' | grep -v '^$' | sed '1d' | cut -c 8- | tr -s ' ' '\n'
23         description = subprocess.run(des, shell=True, capture_output=True, text=True)
24         return description.stdout.strip()
25
26     def generate_version(self, command): #getting the version
27         ver = f"{command} --version | sed -n '1p' | tr -s ' ' | tr '\n' ' '
28         version = subprocess.run(ver, shell=True, capture_output=True, text=True)
29         return version.stdout.strip()
30
31     def generate_related_commands(self, command): #get some related commands :)
32         if command == "chmod":
33             related_comm = f"bash -c 'compgen -c | grep ^ch | head -n 5'"
34         elif command == "rm":
35             related_comm = f"bash -c 'compgen -c | grep ^rm | head -n 5'"
36         elif command == "head":
37             related_comm = f"bash -c 'compgen -c | grep he | head -n 5'"
38         elif command == "apropos":
39             related_comm = f"bash -c 'compgen -c | grep pos | head -n 5'"
40         elif command == "uname":
41             related_comm = f"bash -c 'compgen -c | grep name | head -n 5'"
42         elif command == "sed":
43             related_comm = f"bash -c 'compgen -c | grep se[a-z] | head -n 5'"
44         else:
45             related_comm = f"bash -c 'compgen -c | grep {command} | head -n 5'"
46
47         related = subprocess.run(related_comm, shell=True, capture_output=True, text=True)
```

I also added CommandManual class to get all information about the given command, I made it in functions such as generate_description and generate_syntax and so on to get all needed information about the command. It just saves this information in variables and doesn't actually create the xml file.

```

class XmlSerializer(): #creating the xml file for choosen command
def __init__(self,command):
    self.command=command
def generate_xml(self,command):
    manual= CommandManual(command)
    description = manual.generate_description(command)
    version = manual.generate_version(command)
    related_commands = manual.generate_related_commands(command)
    example = manual.generate_example(command)
    syntax =manual.generate_syntax(command)
    usage=manual.generate_usage_pattern(command)
    doc_link=manual.generate_doc_link(command)

    root = minidom.Document()
    manuals = root.createElement('Manuals')
    root.appendChild(manuals)

    command_manual = root.createElement('CommandManual')
    manuals.appendChild(command_manual)

    command_name = root.createElement('CommandName')
    command_name.appendChild(root.createTextNode(command))
    command_manual.appendChild(command_name)

    command_description = root.createElement('CommandDescription')
    command_description.appendChild(root.createTextNode(description))
    command_manual.appendChild(command_description)

    version_history = root.createElement('VersionHistory')
    version_history.appendChild(root.createTextNode(version))
    command_manual.appendChild(version_history)

```

This is a code snippet from the xml class which takes the command name and make the xml file by the help of CommandManual class

```

class CommandManualVerifier(): #to verify command
def __init__(self):
    pass
def verify_manual(self, command):
    manual_file = f'{command}_manual.xml'

    if not os.path.exists(manual_file):
        print(f'Manual file for command '{command}' does not exist.')
        return

    with open(manual_file, "r") as f:
        content = f.read()

    # Parse XML content
    try:
        xml_content = minidom.parseString(content)
    except Exception as e:
        print(f"Error parsing XML for command '{command}': {str(e)}")
        return

    des = f"man {command} | awk '/^DESCRIPTION/,/^$/' | grep -v '^$' | sed 'Id' | cut -c 8- | tr -s ' '"
    description = subprocess.run(des, shell=True, capture_output=True, text=True)

    ver = f"{command} --version | sed -n '1p' | tr -s ' ' | tr '\n' ' '"
    version = subprocess.run(ver, shell=True, capture_output=True, text=True)

    if command == "chmod":
        related_comm = f"bash -c 'compgen -c | grep ^ch | head -n 5'"
    elif command == "rm":
        related_comm = f"bash -c 'compgen -c | grep ^rm | head -n 5'"

```

This is code snippet from class to verify the xml file, if anything changed it simply shows what changed, if not it prints everything is correct

```

class search_Manual(): #search for / in file
def __init__(self):
    pass

def search_function(self, command, section):
    manual_file = f"{command}_manual.xml"

    if not os.path.exists(manual_file):
        print(f"Manual file for command '{command}' does not exist.")
        return

    with open(manual_file, "r") as f:
        content = f.read()

    # Parse XML content
    try:
        xml_content = minidom.parseString(content)
    except Exception as e:
        print(f"Error parsing XML for command '{command}': {str(e)}")
        return

    print("Command Manual exists !\n")
    print("here is search results:")

    if section == "description":
        command_description = xml_content.getElementsByTagName('CommandDescription')[0].firstChild.data.strip()
        print(command_description)
    elif section == "version":
        version_history = xml_content.getElementsByTagName('VersionHistory')[0].firstChild.data.strip()
        print(version_history)
    elif section == "example":
        example = xml_content.getElementsByTagName('Example')[0].firstChild.data.strip()
        print(example)

```

Search functionality is very clear that user chooses what command to search for, which part exactly, user has 7 options in this case

```

class Recommendation_system():
def __init__(self):
    pass

def Recommendation_function(self, command):
    if command=="cat":
        print("1. tac")
        print("2. less")
        print("3. more")
    elif command == "chmod":
        print("1. chown")
        print("2. chgrp")
    elif command == "wc":
        print("1. sed")
        print("2. awk")
    elif command == "tr":
        print("1. sed")
        print("2. perl")
    elif command == "who":
        print("1. users")
        print("2. whoami")
        print("3. id")
        print("4. w")

```

Recommendation system was done manually, I found it the more suitable way to recommend commands that are very similar in functionality for the ones the user searched for.

Results

```
rana@rana-VirtualBox:~/project2$ /bin/python3 /home/rana/project2/project.py
This is commands list:
cat
chmod
wc
tr
sed
who
cut
date
ls
rm
cp
head
tail
sort
apropos
arch
id
uname
free
cmp
finished generating Manuals !
do you want to verify any command? yes/no
```

After running the code, it displays the list of commands that have been read from file. Then it asks the user to verify any command of his chosen.

```
finished generating Manuals !
do you want to verify any command? yes/no
yes
Enter command you want to verify
cut
Manual content for command 'cut' is correct.
do you want to serch for any command? yes/no
```

This shows a successfully verified command

Now I want to test in case something changed

```
free_manual.xml
1  <?xml version="1.0" ?>
2  <Manuals>
3    <CommandManual>
4      <CommandName>free</CommandName>
5      <CommandDescription>free displays the total amount of free and used physical and swap memory
6      well as the buffers and caches used by the kernel. The information is gathered by parsing
7      /proc/meminfo. The displayed columns are:</CommandDescription>
8      <VersionHistory>free from procs-ng 3.3.17</VersionHistory>
9      <Example>free</Example>
10     <RelatedCommands>fsfreeze
11     e2freefrag
12     free
13     fsfreeze
14     e2freefrag</RelatedCommands>
15     <syntax>free [options]</syntax>
16     <UsagePatterns>Display amount of free and used memory in the system.</UsagePatterns>
17     <OnlineDocumentationLink>https://man7.org/linux/man-pages/man1/free.1.html</OnlineDocumentationLink>
18   </CommandManual>
19 </Manuals>
20
```

This is the original manual for command free

```
free_manual.xml
1  <?xml version="1.0" ?>
2  <Manuals>
3    <CommandManual>
4      <CommandName>free</CommandName>
5      <CommandDescription>fre displays the total amount of free and used physical and swap memory i
6      well as the buffers and caches used by the kernel. The information is gathered by parsing
7      /proc/meminfo. The displayed columns are:</CommandDescription>
8      <VersionHistory>free from procs-ng 3.3.16</VersionHistory>
9      <Example>free</Example>
10     <RelatedCommands>fsfreeze
11     e2freefrag
12     free
13     fsfreeze
14     e2freefrag</RelatedCommands>
15     <syntax>free [options]</syntax>
16     <UsagePatterns>Display amount of free and used memory in the system.</UsagePatterns>
17     <OnlineDocumentationLink>https://man7.org/linux/man-pages/man1/free.1.html</OnlineDocumentationLink>
18   </CommandManual>
19 </Manuals>
20
```

I made these changes circled in red

```
do you want to verify any command? yes/no
yes
Enter command you want to verify
free
Changes detected for command 'free':
Description for command 'free' has changed.
Example for command 'free' has changed.
do you want to serch for any command? yes/no

```

Some search results:

```
Enter command you want to search for
who
Enter what section you want to see (enter a number)
1.description
2.example
3.version
4.related commands
5.Usage Patterns
6.syntax
7.Online Documentation Link
5
Command Manual exists !

here is search results:
Show who is logged on.

-----Recommendation-----
you recently searched for who
you can see this also:
1. users
2. whoami
3. id
4. w
```

```
do you want to serch for any command? yes/no
yes
Enter command you want to search for
cp
Enter what section you want to see (enter a number)
1.description
2.example
3.version
4.related commands
5.Usage Patterns
6.syntax
7.Online Documentation Link
3
Command Manual exists !

here is search results:
cp (GNU coreutils) 8.32

-----Recommendation-----
you recently searched for cp
you can see this also:
1. rsync
2. tar
rana@rana-VirtualBox:~/project2$
```

```
do you want to serch for any command? yes/no
yes
Enter command you want to search for
rm
Enter what section you want to see (enter a number)
1.description
2.example
3.version
4.related commands
5.Usage Patterns
6.syntax
7.Online Documentation Link
7
Command Manual exists !

here is search results:
https://man7.org/linux/man-pages/man1/rm.1.html

-----Recommendation-----
you recently searched for rm
you can see this also:
1. rmdir
2. unlink
3. trash-put
```

```
do you want to serch for any command? yes/no
yes
Enter command you want to search for
id
Enter what section you want to see (enter a number)
1.description
2.example
3.version
4.related commands
5.Usage Patterns
6.syntax
7.Online Documentation Link
1
Command Manual exists !

here is search results:
Print user and group information for each specified USER, or (when USER omitted) for the
current user.

-----Recommendation-----
you recently searched for id
you can see this also:
1. groups
2. whoami
rana@rana-VirtualBox:~/project2$
```


Some of the xml files:

```
1 <?xml version="1.0" ?>
2 <Manuals>
3   <CommandManual>
4     <CommandName>cat</CommandName>
5     <CommandDescription>Concatenate FILE(s) to standard output.</CommandDescription>
6     <VersionHistory>cat (GNU coreutils) 8.32</VersionHistory>
7     <Example>cat file.txt</Example>
8     <RelatedCommands>install-sgmlcatalog
9   </CommandManual>
10 </Manuals>
11 update-ca-certificates
12 tarcat</RelatedCommands>
13   <syntax>cat [options] [file(s)]</syntax>
14   <UsagePatterns>Concatenate and display the content of one or more files.</UsagePatterns>
15   <OnlineDocumentationLink>https://man7.org/linux/man-pages/man1/cat.1.html</OnlineDocumentationLink>
16 </CommandManual>
17 </Manuals>
```

```
1 <?xml version="1.0" ?>
2 <Manuals>
3   <CommandManual>
4     <CommandName>arch</CommandName>
5     <CommandDescription>Print machine architecture.</CommandDescription>
6     <VersionHistory>arch (GNU coreutils) 8.32</VersionHistory>
7     <Example>arch</Example>
8     <RelatedCommands>setarch
9   </CommandManual>
10 apt-ftparchive
11 dpkg-architecture
12 setarch</RelatedCommands>
13   <syntax>arch [options]</syntax>
14   <UsagePatterns>Display machine architecture.</UsagePatterns>
15   <OnlineDocumentationLink>https://man7.org/linux/man-pages/man1/arch.1.html</OnlineDocumentationLink>
16 </CommandManual>
17 </Manuals>
```

```
1 <?xml version="1.0" ?>
2 <Manuals>
3   <CommandManual>
4     <CommandName>rm</CommandName>
5     <CommandDescription>This manual page documents the GNU version of rm. rm removes each specified file. By default, it does not remove directories.</CommandDescription>
6     <VersionHistory>rm (GNU coreutils) 8.32</VersionHistory>
7     <Example>rm test_file</Example>
8     <RelatedCommands>rmt
9   </CommandManual>
10 </Manuals>
11 rmmod
12 rmt-tar
13 rm
14 rmdir</RelatedCommands>
15   <syntax>rm [options] [file(s)]</syntax>
16   <UsagePatterns>Remove files or directories.</UsagePatterns>
17   <OnlineDocumentationLink>https://man7.org/linux/man-pages/man1/rm.1.html</OnlineDocumentationLink>
18 </CommandManual>
19 </Manuals>
```

```

1 <?xml version="1.0" ?>
2 <Manuals>
3   <CommandManual>
4     <CommandName>wc</CommandName>
5     <CommandDescription>Print newline, word, and byte counts for each FILE, and a total line if more than one FILE
6 is specified. A word is a non-zero-length sequence of characters delimited by white space.</CommandDescription>
7     <VersionHistory>wc (GNU coreutils) 8.32</VersionHistory>
8     <Example>cat file.txt | wc -l</Example>
9     <RelatedCommands>pwck
10 iwconfig
11 pwconv
12 shadowconfig
13 hwclock</RelatedCommands>
14     <syntax>wc [options] [file(s)]</syntax>
15     <UsagePatterns>Count the number of lines, words, and bytes in files.</UsagePatterns>
16     <OnlineDocumentationLink>https://man7.org/linux/man-pages/man1/wc.1.html</OnlineDocumentationLink>
17   </CommandManual>
18 </Manuals>

```

```

1 <?xml version="1.0" ?>
2 <Manuals>
3   <CommandManual>
4     <CommandName>tr</CommandName>
5     <CommandDescription>Translate, squeeze, and/or delete characters from standard input, writing to standard out-
6 put.</CommandDescription>
7     <VersionHistory>tr (GNU coreutils) 8.32</VersionHistory>
8     <Example>cat file.txt | tr [a-z] [A-Z]</Example>
9     <RelatedCommands>trap
10 true
11 mkinitramfs
12 pam_extrausers_update
13 fstrim</RelatedCommands>
14     <syntax>tr [options] set1 set2</syntax>
15     <UsagePatterns>Translate or delete characters from standard input, writing to standard output.</UsagePatterns>
16     <OnlineDocumentationLink>https://man7.org/linux/man-pages/man1/tr.1.html</OnlineDocumentationLink>
17   </CommandManual>
18 </Manuals>

```

```

1 <?xml version="1.0" ?>
2 <Manuals>
3   <CommandManual>
4     <CommandName>sort</CommandName>
5     <CommandDescription>Write sorted concatenation of all FILE(s) to standard output.</CommandDescription>
6     <VersionHistory>sort (GNU coreutils) 8.32</VersionHistory>
7     <Example>sort</Example>
8     <RelatedCommands>apt-sortpkgs
9 tsort
10 sort
11 apt-sortpkgs
12 tsort</RelatedCommands>
13     <syntax>sort [options] [file(s)]</syntax>
14     <UsagePatterns>Sort lines of text files.</UsagePatterns>
15     <OnlineDocumentationLink>https://man7.org/linux/man-pages/man1/sort.1.html</OnlineDocumentationLink>
16   </CommandManual>
17 </Manuals>

```

```

1 <?xml version="1.0" ?>
2 <Manuals>
3   <CommandManual>
4     <CommandName>date</CommandName>
5     <CommandDescription>Display the current time in the given FORMAT, or set the system date.</CommandDescription>
6     <VersionHistory>date (GNU coreutils) 8.32</VersionHistory>
7     <Example>date</Example>
8     <RelatedCommands>unix_update
9 update-grub-gfxpayload
10 update-inetd
11 pam_extrausers_update
12 update-gsfontmap</RelatedCommands>
13     <syntax>date [options]</syntax>
14     <UsagePatterns>Display or set the system date and time.</UsagePatterns>
15     <OnlineDocumentationLink>https://man7.org/linux/man-pages/man1/date.1.html</OnlineDocumentationLink>
16   </CommandManual>
17 </Manuals>

```