

## PART 1

In this part we are asked to implement write `execv()` function instead of `execvpe()` function and also support background programs in the shell. When we change `execvpe()` to `execv()` we should specify our path and give `execv()` path as an argument. That's what I did. I resolved the path and give `execv()` function as an argument. For background issue. I just added if it is not background process wait for child if it is background process go directly to prompt that solved my background execution problem.

## PART2

In part 2 redirects are already implemented but I need to give the functionality that is specified in the assignment pdf. Inside the child process for each redirect I gave the permissions which is asked the question then the problem solved.

## PART 3.1

In this part we asked to implement similar function like `xxd` utility. In this function if file provided my code reads from the file if not it takes as `stdin`. What I did is I create variable to determine the size of lines which is 16 bytes. Then I print them in hexadecimal format for given group size but it only works correctly for powers of 2 because assignment pdf says we don't need to consider group sizes that are not powers of 2. You can see some of the outputs below.

```

thrawn@thrawn: ~/Desktop/project/project-1-shell-tyranus
thrawn@thrawn:~/Desktop/project/project-1-shell-tyranus$ ./shellect
thrawn@thrawn:~/home/thrawn/Desktop/project/project-1-shell-tyranus$ Shelllect$ xdd -g 1 input.txt
41 74 20 68 6f 6d 65 2c 20 77 65 20 68 61 76 65
20 68 61 64 20 74 77 65 6c 76 65 20 75 6e 68 61
70 70 79 20 79 65 61 72 73 20 6f 66 20 74 75 72
6d 6f 69 6c 20 61 6e 64 20 64 69 73 73 65 6e 73
69 6f 6e 2c 20 6f 66 20 67 72 6f 75 70 20 63 6f
6e 66 6c 69 63 74 20 61 6e 64 20 63 6c 61 73 73
20 73 74 72 69 66 65 2e 20 4f 66 20 64 69 76 69
73 69 6f 6e 73 2c 20 61 6e 64 20 68 61 74 72 65
64 73 20 61 6e 64 20 61 6e 74 61 67 6f 6e 69 73
6d 73 2e 20 48 61 6c 66 20 61 20 67 65 6e 65 72
61 74 69 6f 6e 20 68 61 73 20 67 72 6f 77 6e 20
75 70 20 6b 6e 6f 77 69 6e 67 20 6e 6f 20 6f 74
68 65 72 20 61 74 6d 6f 73 70 68 65 72 65 2e 20
49 20 62 65 6c 69 65 76 65 20 6f 75 72 20 63 68
69 6c 64 72 65 6e 2c 20 6f 75 72 20 77 68 6f 6c
65 20 63 6f 75 6e 74 72 79 2c 20 63 61 6e 20 61
67 61 69 6e 20 6c 69 76 65 20 69 6e 20 61 20 77
6f 72 6c 64 20 77 68 65 72 65 20 70 65 61 63 65
2c 20 66 72 69 65 6e 64 73 68 69 70 20 61 6e 64
20 6d 75 74 75 61 6c 20 72 65 73 70 65 63 74 2c
20 61 62 69 64 65 2e 0a
thrawn@thrawn:~/home/thrawn/Desktop/project/project-1-shell-tyranus$ Shelllect$ xdd -g 2 input.txt
4174 2068 6f6d 652c 2077 6520 6861 7665
2068 6164 2074 7765 6c76 6520 756e 6861
7070 7920 7965 6172 7320 6f66 2074 7572
6d6f 696c 2061 6e64 2064 6973 7365 6e73
696f 6e2c 206f 6620 6772 6f75 7020 636f
6e66 6c69 6374 2061 6e64 2063 6c61 7373
2073 7472 6966 652e 204f 6620 6469 7669
7369 6f6e 732c 2061 6e64 2068 6174 7265
6473 2061 6e64 2061 6e74 6167 6f6e 6973
6d73 2e20 4861 6c66 2061 2067 656e 6572
6174 696f 6e20 6861 7320 6772 6f77 6e20
7570 206b 6e6f 7769 6e67 206e 6f20 6f74
6865 7220 6174 6d6f 7370 6865 7265 2e20
4920 6265 6c69 6576 6520 6f75 7220 6368
696c 6472 656e 2c20 6f75 7220 7768 6f6c
6520 636f 756e 7472 792c 2063 616e 2061
6761 696e 206c 6976 6520 696e 2061 2077
6f72 6c64 2077 6865 7265 2070 6561 6365
2c20 6672 6965 6e64 7368 6970 2061 6e64
206d 7574 7561 6c20 7265 7370 6563 742c
2061 6269 6465 2e0a
thrawn@thrawn:~/home/thrawn/Desktop/project/project-1-shell-tyranus$ Shelllect$

```

```

thrawn@thrawn:/home/thrawn/Desktop/project/project-1-shell-tyranus Shellec$ xdd -g 8 input.txt
417420686f6d652c 2077652068617665
2068616420747765 6c766520756e6861
7070792079656172 73206f6620747572
6d6f696c20616e64 2064697373656e73
696f6e2c206f6620 67726f757020636f
6e666c6963742061 6e6420636c617373
207374726966652e 204f662064697669
73696f6e732c2061 6e64206861747265
647320616e642061 6e7461676f6e6973
6d732e2048616c66 20612067656e6572
6174696f6e206861 732067726f776e20
7570206b0e6f7769 6e67206e6f206f74
6865722061746d6f 7370686572652e20
492062656c696576 65206f7572206368
696c6472656e2c20 6f75722077686f6c
6520636f756e7472 792c2063616e2061
6761696e206c6976 6520696e20612077
6f726c6420776865 7265207065616365
2c20667269656e64 7368697020616e64
206d757475616c20 726573706563742c
2061626964652e8a

```

## PART 3.2

In this part we asked to implement alias function which saves shortcuts for commonly used commands. Since there is rule that says aliases shouldn't disappear once we close the shell. For this problem I thought I create file for aliases and save them inside therefore they won't disappear when we close the shell. So when the alias command type I write the alias name and the command in a file with coma separated. Then I make and if statement if the alias name is called execute the related command. My alias command also supports if the same alias name created twice it updates it self to last one. Also instead of providing inside the string I preferred to just type it because I think typing directly much faster than typing in the string when we creating our alias. Here is some outputs of my command.

```

thrawn@thrawn:/home/thrawn/Desktop/project/project-1-shell-tyranus Shellec$ alias short ls
alias 'short' added.
thrawn@thrawn:/home/thrawn/Desktop/project/project-1-shell-tyranus Shellec$ short
alias_file append.txt arnold_wakey.mp3 build CMakeLists.txt custom.txt input.txt Makefile module output.txt README.md shellec src
thrawn@thrawn:/home/thrawn/Desktop/project/project-1-shell-tyranus Shellec$ alias com pwd
alias 'com' added.
thrawn@thrawn:/home/thrawn/Desktop/project/project-1-shell-tyranus Shellec$ com
/home/thrawn/Desktop/project/project-1-shell-tyranus
thrawn@thrawn:/home/thrawn/Desktop/project/project-1-shell-tyranus Shellec$ alias short ls -lha
alias 'short' added.
thrawn@thrawn:/home/thrawn/Desktop/project/project-1-shell-tyranus Shellec$ short
total 1,3M
drwxrwxr-x 8 thrawn thrawn 4,0K Kas 12 19:14 .
drwxrwxr-x 3 thrawn thrawn 4,0K Kas 8 21:56 ..
-rw-r--r-- 1 thrawn thrawn 55 Kas 12 19:58 alias_file
-rw-r--r-- 1 thrawn thrawn 471 Kas 11 18:24 append.txt
-rw-r--r-- 1 thrawn thrawn 1,2M Kas 10 02:11 arnold_wakey.mp3
drwxrwxr-x 3 thrawn thrawn 4,0K Kas 12 19:14 build
-rw-r--r-- 1 thrawn thrawn 3,3K Kas 8 21:56 .clang-format
-rw-r--r-- 1 thrawn thrawn 486 Kas 8 21:56 CMakeLists.txt
-rw-r--r-- 1 thrawn thrawn 40 Kas 11 21:41 custom.txt
drwxrwxr-x 8 thrawn thrawn 4,0K Kas 12 18:23 .git
drwxrwxr-x 4 thrawn thrawn 4,0K Kas 8 21:56 .github
-rw-r--r-- 1 thrawn thrawn 568 Kas 8 21:56 .gitignore
-rw-r--r-- 1 thrawn thrawn 328 Kas 10 01:54 input.txt
-rw-r--r-- 1 thrawn thrawn 1,4K Kas 8 21:56 Makefile
drwxrwxr-x 2 thrawn thrawn 4,0K Kas 12 18:46 module
-rw-r--r-- 1 thrawn thrawn 350 Kas 11 18:29 output.txt
-rw-r--r-- 1 thrawn thrawn 529 Kas 8 21:56 README.md
drwxrwxr-x 1 thrawn thrawn 31K Kas 12 19:14 shellec
drwxrwxr-x 2 thrawn thrawn 4,0K Kas 8 21:56 src
drwxrwxr-x 2 thrawn thrawn 4,0K Kas 12 15:54 .vscode
thrawn@thrawn:/home/thrawn/Desktop/project/project-1-shell-tyranus Shellec$

```

## PART 3.3

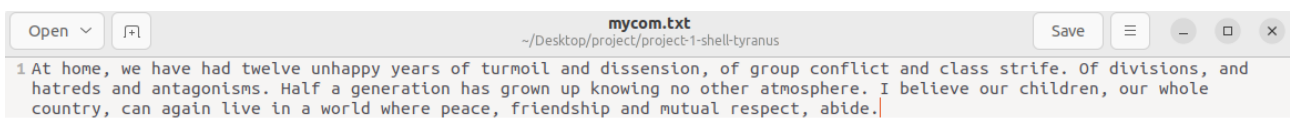
In this part we asked to implement good\_morning alarm so we are providing number of minutes and path of the music file then it plays our music file after given number of minutes. In this problem I used mpg123. So I created a cron job since cron works when write in the kernel I just create cron job with definin a cron job string and giving my path and specifying mpg123. Then I make a system call in my code and it worked. I can't show you sound but you can try it. It also prints if alarm is scheduled. Here is my command and it's output.

```
thrawn@thrawn:/home/thrawn/Desktop/project/project-1-shell-tyranus Shellest$ good_morning 1 /home/thrawn/Music/arnold_wakey.mp3
Alarm scheduled to play the song in 1 minutes.
thrawn@thrawn:/home/thrawn/Desktop/project/project-1-shell-tyranus Shellest$ █
```

## PART 3.4

In this part I implemented my builtin command. My command called “all\_occurrences”. So my function takes 3 inputs. First one is either “-c” or “-d”. “-c” for specifying count “-d” for specifying delete. Second argument is a word and the last argument is a file. So if the specifier is -c it counts how many times is this word occurred in the file. If specifier -d it deletes all occurrences in the file. Here are some outputs.

This is the txt file that I will use:



```
1 At home, we have had twelve unhappy years of turmoil and dissension, of group conflict and class strife. Of divisions, and
  hatreds and antagonisms. Half a generation has grown up knowing no other atmosphere. I believe our children, our whole
  country, can again live in a world where peace, friendship and mutual respect, abide.
```

Here is a count example. As you can see it correctly counted the number of occurrences.

```
thrawn@thrawn:/home/thrawn/Desktop/project/project-1-shell-tyranus Shellest$ all_occurrences -c we mycom.txt
The word 'we' occurred 1 times in the file 'mycom.txt'.
thrawn@thrawn:/home/thrawn/Desktop/project/project-1-shell-tyranus Shellest$ all_occurrences -c and mycom.txt
The word 'and' occurred 5 times in the file 'mycom.txt'.
thrawn@thrawn:/home/thrawn/Desktop/project/project-1-shell-tyranus Shellest$ █
```

Here is delete example and file after this execution.

```
thrawn@thrawn:/home/thrawn/Desktop/project/project-1-shell-tyranus Shellest$ all_occurrences -d and mycom.txt
thrawn@thrawn:/home/thrawn/Desktop/project/project-1-shell-tyranus Shellest$ cat mycom.txt
At home, we have had twelve unhappy years of turmoil dissension, of group conflict class strife. Of divisions, hatreds antagonisms. Half a generation has grown up knowing no other
atmosphere. I believe our children, our whole country, can again live in a world where peace, friendship mutual respect, abide.
thrawn@thrawn:/home/thrawn/Desktop/project/project-1-shell-tyranus Shellest$ █
```

## PART 4

In this part we are implementing a kernel module. First I was getting error but after some research I learned I had to disable Secure Boot from BIOS on my computer. After that I could able execute the model. Since we need to be a super user in this problem I used sudo command. For example I used sudo dmesg to check the kernel log. So what we need to in this part is to given a pid we need to plot it is subtree. So I first started printing this subtree in the kernel log. So first I implemented my function with only pid argument. For example you can see the sub tree of pid = 1193 on the kernel log below.

```
thrawn@thrawn: ~/Desktop/project/project-1-shell-tyranus
thrawn@thrawn:~/Desktop/project/project-1-shell-tyranus$ ./shellect
thrawn@thrawn:/home/thrawn/Desktop/project/project-1-shell-tyranus Shelllect$ psvis 1193
[sudo] password for thrawn:
thrawn@thrawn:/home/thrawn/Desktop/project/project-1-shell-tyranus Shelllect$ sudo dmesg
```

```
[26027.211411] Parent Process: systemd[1193]
[26027.211414] [1193] systemd, Creation Time: 26071.181894213
[26027.211416] [1195] (sd-pam), Creation Time: 26071.477488565
[26027.211418] [1214] pipewire, Creation Time: 26072.796448045
[26027.211419] [1215] pipewire-media-, Creation Time: 26072.797525070
[26027.211421] [1216] pulseaudio, Creation Time: 26072.798246054
[26027.211422] [1305] dbus-daemon, Creation Time: 26076.061119890
[26027.211423] [1398] xdg-document-po, Creation Time: 26081.434440166
[26027.211425] [1401] xdg-permission-, Creation Time: 26081.803764397
[26027.211426] [1423] gvfsd, Creation Time: 26084.065868621
[26027.211427] [2085] gvfsd-trash, Creation Time: 26110.524040275
[26027.211429] [19917] gvfsd-http, Creation Time: 46698.188884640
[26027.211430] [1438] gvfsd-fuse, Creation Time: 26084.466749266
[26027.211432] [1583] snapd-desktop-i, Creation Time: 26093.713826385
[26027.211433] [1676] snapd-desktop-i, Creation Time: 26093.856314579
[26027.211434] [1590] tracker-miner-f, Creation Time: 26093.736970466
[26027.211435] [1706] gvfs-udisks2-vo, Creation Time: 26095.353112332
[26027.211436] [1717] gvfs-gphoto2-vo, Creation Time: 26095.427689669
[26027.211437] [1721] gvfs-mtp-volume, Creation Time: 26095.582308411
[26027.211439] [1726] gvfs-afc-volume, Creation Time: 26095.675814341
[26027.211440] [1757] gvfs-goa-volume, Creation Time: 26095.891635072
[26027.211441] [1843] at-spi-bus-laun, Creation Time: 26098.992491187
[26027.211442] [1849] dbus-daemon, Creation Time: 26099.085473989
[26027.211443] [1772] goa-daemon, Creation Time: 26096.024078397
[26027.211445] [1857] goa-identity-se, Creation Time: 26099.514852006
[26027.211446] [1896] gnome-session-c, Creation Time: 26100.153640619
[26027.211447] [1909] xdg-desktop-por, Creation Time: 26100.413909145
[26027.211449] [1912] gnome-session-b, Creation Time: 26100.461288021
[26027.211450] [2173] evolution-alarm, Creation Time: 26111.967573934
[26027.211451] [2184] gsd-disk-utilit, Creation Time: 26111.975029065
[26027.211452] [3716] update-notifier, Creation Time: 26172.344006249
[26027.211453] [1922] xdg-desktop-por, Creation Time: 26100.958995817
[26027.211454] [1939] gnome-shell, Creation Time: 26101.159429849
[26027.211455] [2504] gjs, Creation Time: 26116.963801401
[26027.211457] [3346] firefox, Creation Time: 26154.389659382
[26027.211458] [7920] spotify, Creation Time: 32288.777970652
[26027.211459] [8007] spotify, Creation Time: 32291.812475790
[26027.211460] [8037] spotify, Creation Time: 32293.284339532
[26027.211461] [8008] spotify, Creation Time: 32291.828242539
[26027.211463] [8053] spotify, Creation Time: 32295.291357633
[26027.211464] [8081] spotify, Creation Time: 32296.583856640
[26027.211465] [19883] gnome-control-c, Creation Time: 46668.483008696
[26027.211467] [19997] oosplash, Creation Time: 46871.320998899
[26027.211468] [1978] xdg-desktop-por, Creation Time: 26103.689736358
[26027.211469] [1983] at-spi2-registr, Creation Time: 26103.865667819
[26027.211470] [2061] dconf-service, Creation Time: 26109.372474231
[26027.211471] [2062] evolution-sourc, Creation Time: 26109.410291487
[26027.211472] [2081] evolution-calen, Creation Time: 26110.427247125
[26027.211474] [2051] gnome-shell-cal, Creation Time: 26108.599114370
[26027.211475] [2120] evolution-addre, Creation Time: 26111.577138642
[26027.211476] [2102] gjs, Creation Time: 26111.552759240
[26027.211478] [2130] sh, Creation Time: 26111.808881750
[26027.211479] [2134] ibus-daemon, Creation Time: 26111.810414235
[26027.211480] [2321] ibus-memconf, Creation Time: 26113.128369056
[26027.211481] [2322] ibus-extension-, Creation Time: 26113.212153876
[26027.211482] [2359] ibus-engine-sim, Creation Time: 26114.093023522
```

As you can see I can show the parents and child but I couldn't print a graph of this using gnu plot. But in the kernel log is quite obvious if it is printed with a space previous process is parent process of this process. When you type my kernel psvis <pid> and go to kernel log by sudo dmesg you can check it.