EXEC calls to overlay process images

✓Exec calls completely overlay the image of the process where they "are called and the control would not return to the point of next instruction after the exec call.

✓Execl used in the first example – l is for list where the complete list of arguments should be statically specified in the code. The path of the commands also have to be absolutely identified. This path location is automatic with the usage of execlp. (code usage shown in the next example)

✓Limitation of execl and execlp are that the arguments are having to be statically specified in the function call itself.

Execlp example

```
# include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main ()
{
pid_t pid;   // this is to use the pid data type – relevant headers
    above
pid = fork();
if (pid == 0)
execlp("ls,"ls", NULL);  // child image is now ls command
else
{
wait (NULL);   // parent waits for the child to complete execn.
printf("Parent Process gets the control \n");
printf ("Parent Has waited for Child to Complete");
}
}
```

Note: path is automatically detected searching the PATH var

Exec variants syntax

```
char *const argv[] = {"/bin/ls",“ls”,"-l", NULL};
execv(argv[0], argv);
```

Note: arguments are stored in an array which is what you passed statically in the call with execl.V stands for vector

Path needs to be specified which is the first arguments and the rest are treated as arguments to the command NULL terminaed.

```
char *const cmd[] = {"ls",“ls”,"-l", NULL};
execvp(cmd[0], cmd);
```

Combines the features of both v and p calls.

```
char *args[] = {"ls", "-aF", "/", 0};
char *env[] = { 0 }; /* leave the environment list null */
execve("/bin/ls", args, env);
```

```
int ret;
char *cmd[] = { "ls", "-l", (char *)0 };
char *env[] = { "HOME=/usr/home", "LOGNAME=home", (char
   *)0 };
ret = execve ("/bin/ls", cmd, env);
```

These are some good online resources on exec calls and its variants explainataion

- https://linuxhint.com/exec_linux_system_call_c/
- https://www.cs.rutgers.edu/~pxk/416/notes/c-tutorials/exec.html
- https://www.oreilly.com/library/view/secure-programming-cookbook/0596003943/ch01s07.html