

# Author

---

2024011240 차호련

## exec

---

Turns a current process to whole different process.

Still, the graph of it will not change.

Which means if it has parent process, the parent will still think it is his children.

The turned process(newly executing) will preserve its own process id.

After called, since process is now target programme found, existing process will not continue.

Which means no code after exec-like functions must execute, otherwise stands for an exception situation.

## exec-like functions naming

- exec

implements the work of `exec`.

- l

Using variable argument (aka `va_list`) for sending arguments for new process.

- v

Using dynamic vector for sending arguments for new process.

Sends 0(aka null) for a last index, which could notify that is the last element of the argument list.

- e

Environment variables could be sent. Explicitly.

- v

The position of the process will be searched with `PATH`.

## Measuring

---

- Execute the `spawn` written in a book of `exec`.
- Measure the time `spawn` taking.

## function `spawn`

- Creates a new process (fork)

- Child process (newly created) will turn into `ls`. So we will no require the host process to be turned.

## function `main`

- call `spawn` five times.
- 

I am iterating the spawn function five times.

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
#include <wait.h>
#include <time.h>

/* Spawn a child process running a new program. PROGRAM is the name of the program
to run; the path will be searched for this program. ARG_LIST is a NULL-terminated
list of character strings to be passed as the program's argument list. Returns the
process ID of the spawned process. */
int spawn(const char* program, char** const arg_list)
{
    pid_t child_pid;

    /* Duplicate this process. */
    child_pid = fork();
    if (child_pid != 0)
        /* This is the parent process. */
        return child_pid;

    else {
        /* Now execute PROGRAM, searching for it in the path. */
        execvp(program, arg_list);

        /* The execvp function returns only if an error occurs. */
        perror("an error occurred in execvp\n");
        abort();
    }
}

int main()
{
    FILE* _ = fopen("_a", "w");

    clock_t
        global_time = clock(),
        global_time_end,
        loc_time = clock(),
        loc_time_end;

    /* The argument list to pass to the "ls" command. */
```

```
char* arg_list[] = {
    "ls", /* argv[0], the name of the program. */
    "-l",
    "/",
    NULL /* The argument list must end with a NULL. */
};

for(int i = 0; i < 5; i++) {
    loc_time = clock();
    /* Spawn a child process running the "ls" command. Ignore the
    returned child process ID. */
    spawn("ls", arg_list);

    wait(0); // wait until child process die

    loc_time_end = clock();
    printf("loc time %d, %d\n", i, loc_time_end - loc_time);
    fprintf(_, "loc time %d, %d\n", i, loc_time_end - loc_time);
}

global_time_end = clock();
printf("done with main program, elapsed global: %d\n", global_time_end -
global_time);
fprintf(_, "done with main program, elapsed global: %d\n", global_time_end -
global_time);

fclose(_);
return 0;
}
```

## Output stored in console

---

The whole output will be printed out on stdout.  
Including the output of ls, and time measurement.

```
ae2f@fedora:~/Desktop/assignment$ gcc ./a.c && ./a.out
total 20
dr-xr-xr-x.  1 root root    0 Jul 17 09:00 afs
lrwxrwxrwx.  1 root root    7 Jul 17 09:00 bin -> usr/bin
dr-xr-xr-x.  6 root root 4096 Nov 10 12:36 boot
drwxr-xr-x. 21 root root 4280 Nov 11 21:42 dev
drwxr-xr-x.  1 root root 4834 Nov 10 16:54 etc
drwxr-xr-x.  1 root root    8 Nov 10 12:15 home
lrwxrwxrwx.  1 root root    7 Jul 17 09:00 lib -> usr/lib
lrwxrwxrwx.  1 root root    9 Jul 17 09:00 lib64 -> usr/lib64
drwx-----. 1 root root    0 Oct 24 23:47 lost+found
drwxr-xr-x.  1 root root    0 Jul 17 09:00 media
drwxr-xr-x.  1 root root    0 Jul 17 09:00 mnt
drwxr-xr-x.  1 root root   44 Nov 10 14:46 opt
dr-xr-xr-x. 575 root root    0 Nov 11 21:42 proc
dr-xr-x---.  1 root root  226 Nov 10 14:46 root
drwxr-xr-x.  57 root root 1420 Nov 11 21:44 run
lrwxrwxrwx.  1 root root    8 Jul 17 09:00 sbin -> usr/sbin
drwxrwxrwx.  1 root root    8 Jul 17 09:00 sbin -> usr/sbin
drwxr-xr-x.  1 root root    0 Jul 17 09:00 srv
dr-xr-xr-x. 13 root root    0 Nov 11 21:42 sys
drwxrwxrwt. 28 root root  660 Nov 12 02:22 tmp
drwxr-xr-x.  1 root root  168 Oct 24 23:49 usr
drwxr-xr-x.  1 root root  258 Nov 10 13:43 var
loc time 0, 186
total 20
dr-xr-xr-x.  1 root root    0 Jul 17 09:00 afs
lrwxrwxrwx.  1 root root    7 Jul 17 09:00 bin -> usr/bin
dr-xr-xr-x.  6 root root 4096 Nov 10 12:36 boot
drwxr-xr-x. 21 root root 4280 Nov 11 21:42 dev
drwxr-xr-x.  1 root root 4834 Nov 10 16:54 etc
drwxr-xr-x.  1 root root    8 Nov 10 12:15 home
lrwxrwxrwx.  1 root root    7 Jul 17 09:00 lib -> usr/lib
lrwxrwxrwx.  1 root root    9 Jul 17 09:00 lib64 -> usr/lib64
drwx-----. 1 root root    0 Oct 24 23:47 lost+found
drwxr-xr-x.  1 root root    0 Jul 17 09:00 media
```

```
drwxr-xr-x.  1 root root    0 Jul 17 09:00 mnt
drwxr-xr-x.  1 root root   44 Nov 10 14:46 opt
dr-xr-xr-x. 575 root root    0 Nov 11 21:42 proc
dr-xr-x---.  1 root root  226 Nov 10 14:46 root
drwxr-xr-x.  57 root root 1420 Nov 11 21:44 run
lrwxrwxrwx.  1 root root    8 Jul 17 09:00 sbin -> usr/sbin
drwxr-xr-x.  1 root root    0 Jul 17 09:00 srv
dr-xr-xr-x.  13 root root    0 Nov 11 21:42 sys
drwxrwxrwt.  28 root root  660 Nov 12 02:22 tmp
drwxr-xr-x.  1 root root  168 Oct 24 23:49 usr
drwxr-xr-x.  1 root root  258 Nov 10 13:43 var
loc time 1, 186
total 20
dr-xr-xr-x.  1 root root    0 Jul 17 09:00 afs
lrwxrwxrwx.  1 root root    7 Jul 17 09:00 bin -> usr/bin
dr-xr-xr-x.  6 root root 4096 Nov 10 12:36 boot
drwxr-xr-x.  21 root root 4280 Nov 11 21:42 dev
drwxr-xr-x.  1 root root 4834 Nov 10 16:54 etc
```

```
drwxr-xr-x.  1 root root    8 Nov 10 12:15 home
lrwxrwxrwx.  1 root root    7 Jul 17 09:00 lib -> usr/lib
lrwxrwxrwx.  1 root root    9 Jul 17 09:00 lib64 -> usr/lib64
drwx-----. 1 root root    0 Oct 24 23:47 lost+found
drwxr-xr-x.  1 root root    0 Jul 17 09:00 media
drwxr-xr-x.  1 root root    0 Jul 17 09:00 mnt
drwxr-xr-x.  1 root root   44 Nov 10 14:46 opt
dr-xr-xr-x. 575 root root    0 Nov 11 21:42 proc
dr-xr-x---.  1 root root  226 Nov 10 14:46 root
drwxr-xr-x.  57 root root 1420 Nov 11 21:44 run
lrwxrwxrwx.  1 root root    8 Jul 17 09:00 sbin -> usr/sbin
drwxr-xr-x.  1 root root    0 Jul 17 09:00 srv
dr-xr-xr-x.  13 root root    0 Nov 11 21:42 sys
drwxrwxrwt.  28 root root  660 Nov 12 02:22 tmp
drwxr-xr-x.  1 root root  168 Oct 24 23:49 usr
drwxr-xr-x.  1 root root  258 Nov 10 13:43 var
loc time 2, 183
total 20
```

```

drwxr-xr-x.  1 root root  258 Nov 10 13:43 var
loc time 2, 183
total 20
dr-xr-xr-x.  1 root root    0 Jul 17 09:00 afs
lrwxrwxrwx.  1 root root    7 Jul 17 09:00 bin -> usr/bin
dr-xr-xr-x.  6 root root 4096 Nov 10 12:36 boot
drwxr-xr-x. 21 root root 4280 Nov 11 21:42 dev
drwxr-xr-x.  1 root root 4834 Nov 10 16:54 etc
drwxr-xr-x.  1 root root    8 Nov 10 12:15 home
lrwxrwxrwx.  1 root root    7 Jul 17 09:00 lib -> usr/lib
lrwxrwxrwx.  1 root root    9 Jul 17 09:00 lib64 -> usr/lib64
drwx-----.  1 root root    0 Oct 24 23:47 lost+found
drwxr-xr-x.  1 root root    0 Jul 17 09:00 media
drwxr-xr-x.  1 root root    0 Jul 17 09:00 mnt
drwxr-xr-x.  1 root root   44 Nov 10 14:46 opt
dr-xr-xr-x. 575 root root    0 Nov 11 21:42 proc
dr-xr-x---.  1 root root  226 Nov 10 14:46 root
drwxr-xr-x.  57 root root 1420 Nov 11 21:44 run

```

```

drwxr-xr-x.  57 root root 1420 Nov 11 21:44 run
lrwxrwxrwx.  1 root root    8 Jul 17 09:00 sbin -> usr/sbin
drwxr-xr-x.  1 root root    0 Jul 17 09:00 srv
dr-xr-xr-x. 13 root root    0 Nov 11 21:42 sys
drwxrwxrwt. 28 root root  660 Nov 12 02:22 tmp
drwxr-xr-x.  1 root root  168 Oct 24 23:49 usr
drwxr-xr-x.  1 root root  258 Nov 10 13:43 var
loc time 3, 198
total 20
dr-xr-xr-x.  1 root root    0 Jul 17 09:00 afs
lrwxrwxrwx.  1 root root    7 Jul 17 09:00 bin -> usr/bin
dr-xr-xr-x.  6 root root 4096 Nov 10 12:36 boot
drwxr-xr-x. 21 root root 4280 Nov 11 21:42 dev
drwxr-xr-x.  1 root root 4834 Nov 10 16:54 etc
drwxr-xr-x.  1 root root    8 Nov 10 12:15 home
lrwxrwxrwx.  1 root root    7 Jul 17 09:00 lib -> usr/lib
lrwxrwxrwx.  1 root root    9 Jul 17 09:00 lib64 -> usr/lib64
drwx-----.  1 root root    0 Oct 24 23:47 lost+found

```

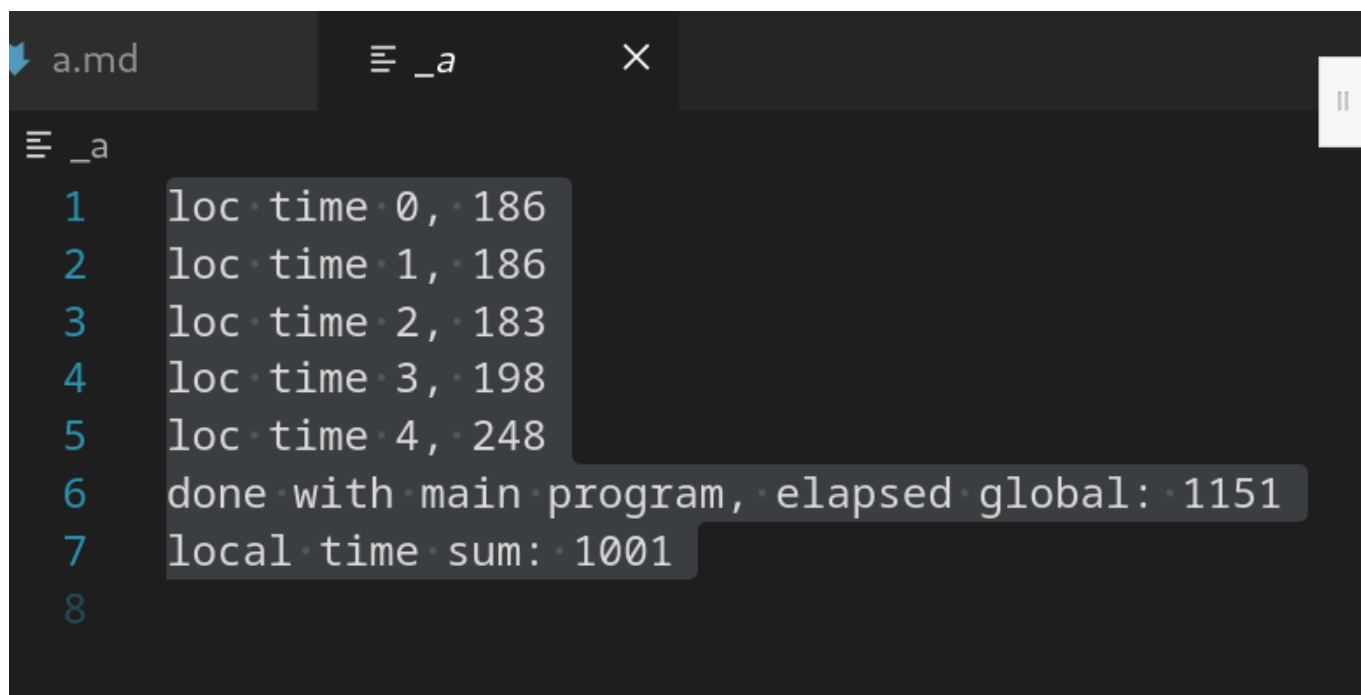
```
lrwxrwxrwx.    1 root root    7 Jul 17 09:00 bin -> usr/bin
dr-xr-xr-x.    6 root root 4096 Nov 10 12:36 boot
drwxr-xr-x.   21 root root 4280 Nov 11 21:42 dev
drwxr-xr-x.    1 root root 4834 Nov 10 16:54 etc
drwxr-xr-x.    1 root root    8 Nov 10 12:15 home
lrwxrwxrwx.    1 root root    7 Jul 17 09:00 lib -> usr/lib
lrwxrwxrwx.    1 root root    9 Jul 17 09:00 lib64 -> usr/lib64
drwx-----.    1 root root    0 Oct 24 23:47 lost+found
drwxr-xr-x.    1 root root    0 Jul 17 09:00 media
drwxr-xr-x.    1 root root    0 Jul 17 09:00 mnt
drwxr-xr-x.    1 root root   44 Nov 10 14:46 opt
dr-xr-xr-x.  575 root root    0 Nov 11 21:42 proc
dr-xr-x---.    1 root root  226 Nov 10 14:46 root
drwxr-xr-x.   57 root root 1420 Nov 11 21:44 run
lrwxrwxrwx.    1 root root    8 Jul 17 09:00 sbin -> usr/sbin
drwxr-xr-x.    1 root root    0 Jul 17 09:00 srv
dr-xr-xr-x.   13 root root    0 Nov 11 21:42 sys
drwxrwxrwt.   28 root root  660 Nov 12 02:22 tmp
```

```
lrwxrwxrwx.    1 root root    9 Jul 17 09:00 lib64 -> usr/lib64
drwx-----.    1 root root    0 Oct 24 23:47 lost+found
drwxr-xr-x.    1 root root    0 Jul 17 09:00 media
drwxr-xr-x.    1 root root    0 Jul 17 09:00 mnt
drwxr-xr-x.    1 root root   44 Nov 10 14:46 opt
dr-xr-xr-x.  575 root root    0 Nov 11 21:42 proc
dr-xr-x---.    1 root root  226 Nov 10 14:46 root
drwxr-xr-x.   57 root root 1420 Nov 11 21:44 run
lrwxrwxrwx.    1 root root    8 Jul 17 09:00 sbin -> usr/sbin
drwxr-xr-x.    1 root root    0 Jul 17 09:00 srv
dr-xr-xr-x.   13 root root    0 Nov 11 21:42 sys
drwxrwxrwt.   28 root root  660 Nov 12 02:22 tmp
drwxr-xr-x.    1 root root  168 Oct 24 23:49 usr
drwxr-xr-x.    1 root root  258 Nov 10 13:43 var
loc time 4, 248
done with main program, elapsed global: 1151
local time sum: 1001
```

```
ae2f@fedora:~/Desktop/assignment$
```

Output stored in file "\_a"

Time measurement will be stored in a file called `_a` for my convenience of checking them at once.  
Content of file will be printed on console also.



```
a.md  _a  X  || c
≡ _a
1  loc time 0, 186
2  loc time 1, 186
3  loc time 2, 183
4  loc time 3, 198
5  loc time 4, 248
6  done with main program, elapsed global: 1151
7  local time sum: 1001
8
```

It seems

## Graph Drawing

I am going to use python for drawing a graph.  
So it could be checked both on Linux and Windows.

```
import matplotlib.pyplot as plt

x = [0, 1, 2, 3, 4]
y = [186, 186, 183, 198, 248]

plt.plot(x, y, marker='o', linestyle='--', color='b')

plt.title('Execution time measurement')
plt.xlabel('Execution Index')
plt.ylabel('Execution Time')

plt.legend()

plt.grid()
plt.show()
```

## Figure Screenshot



