

EXEC FUNCTIONS

- Exec replaces the calling process by a new program
- The new program has same process ID as the calling process
- No new program is created , exec just replaces the current process by a new program

```
#include <unistd.h>

int execl ( const char *pathname, const char *arg0 ,... /*(char *) 0*/);
int execv (const char *pathname, char * const argv[ ]);
int execl (const char *pathname, const char *arg0 ,... /*(char *) 0,
          char *const envp[ ] */);

int execve ( const char *pathname, char *const argv[ ], char *const envp [ ]);
int execlp (const char *filename, const char *arg0 ,... /*(char *) 0*/);
int execvp (const char *filename ,char *const argv[ ]);
```

```
#include <sys/types.h>
#include <sys/wait.h>
#include "ourhdr.h"
char *env_init[ ] =
{ "USER=unknown", "PATH=/tmp", NULL };
int main(void)
{
    pid_t pid;
    if ( (pid = fork()) < 0)
        err_sys("fork error");
    else if (pid == 0) {
        /* specify pathname, specify environment */
        if ( execl ("/home/stevens/bin/echoall",
```

```

"echoall", "myarg1", "MY ARG2",
(char *) 0, env_init) < 0)
    err_sys("execle error");
}
if (waitpid(pid, NULL, 0) < 0)
    err_sys("wait error");

if ( (pid = fork()) < 0)
    err_sys("fork error");

else if (pid == 0) {
/* specify filename, inherit environment */
    if (execlp("echoall",
              "echoall", "only 1 arg",
              (char *) 0) < 0)
        err_sys("execlp error");
    }
    exit(0);
}

```

Changing user IDs and group IDs

- Prototype

```
#include <sys/types.h>
```

```
#include <unistd.h>
```

```
int setuid (uid_t uid);
```

```
int setgid (gid_t gid);
```

- Rules

1. If the process has superuser privilege, the setuid function sets – real user ID, effective user ID , saved set-user-ID to uid
2. If the process doesnot have superuser privilege, but uid equals either real user ID or saved set-user-ID, setuid sets only effective user ID to uid
3. If neither of the two conditions is true, errno is set to EPERM and an error is returned

ID	exec	exec
	Set-user-ID bit off	Set-user-Id bit on
Real user ID	unchanged	unchanged
Effective user ID	unchanged	Set from user ID of program file
Saved set user ID	copied from effective user ID	copied from effective user ID

ID	Superuser	Unprivileged user
Real user ID	Set to uid	unchanged
Effective user ID	Set to uid	Set to uid
Saved set-user ID	Set to uid	unchanged

Source : <http://elearningatria.files.wordpress.com/2013/10/cse-iv-unix-and-shell-programming-10cs44-notes.pdf>