

# Lab Shell

---

Dylan Alvarez, 98225

1º cuatrimestre, 20/04/2018

## Parte 1: Invocación de comandos

---

### Preguntas

- *Pregunta:* ¿cuáles son las diferencias entre la *syscall* `execve(2)` y la familia de *wrappers* proporcionados por la librería estándar de C (*libc*) `exec(3)` ?
- *Respuesta:* La *syscall* `execve(2)` toma la ruta absoluta del archivo ejecutable, mientras que los *wrappers* de la librería estándar permiten resolver rutas relativas, si la ruta dada no comienza con `'/'`. Además, otorgan firmas de función alternativas para facilitar usos comunes.

### Código modificado para la parte 1

exec.c

```
void exec_cmd(struct cmd *cmd) {
    switch (cmd->type) {
        case EXEC:
            // spawns a command
            if (execvp(
                ((struct execcmd *) cmd)->argv[0],
                ((struct execcmd *) cmd)->argv
            ) == -1) {
                perror(SHELL_NAME);
            }
            break;
    }
}
```

parsing.c

```
// this function will be called for every token, and it should
// expand environment variables. In other words, if the token
// happens to start with '$', the correct substitution with the
// environment value should be performed. Otherwise the same
// token is returned.
//
// Hints:
// - check if the first byte of the argument
//   contains the '$'
// - expand it and copy the value
//   to 'arg'
static char *expand_env_var(char *arg) {
    char *env;
    if (arg[0] == '$' && (env = getenv(&arg[1])) != NULL)
        strncpy(arg, env, ARGSIZE);
        arg[ARGSIZE - 1] = 0;
    return arg;
}
```