

Apellidos .....

Nombre .....

Nº de matrícula.....

**Examen de prácticas****Tiempo Estimado 20 m**

```
#include ...

int extern errno;

int pipes[10][2]; //matriz para albergar los descriptores de hasta 11 pipes distintos

void trataError (int status) {
    if (status == -1) {
        fprintf (stderr, "error %d, %s\n", errno, strerror (errno));
        exit (1);
    }
}

void visualizaPrompt () {
    fprintf (stderr, "msh$ ");
}

int main () {
    char *fichEnt = NULL;
    int i, j, miPipe[2], numCmds, pid, status;

    while (!esFinFichero ()) {           //mientras no sea fin de fichero
        visualizaPrompt (); //msh$
        numCmds = leeYanaliza ();        //lee y analiza sintáctica mente
        if (numCmds > 0) {               //si no ha habido error sintáctico

            for (i = 0; i < (numCmds - 1); i++) {
                trataError (pipe (miPipe));
                pipes[i][0] = miPipe[0];
                pipes[i][1] = miPipe[1];
            }

            for (i = 0; i < numCmds; i++) {
                trataError (pid = fork ());
                if (pid != 0) { //padre
                    } else { //hijo

                        if ((i == 0) && ((fichEnt = obtenFichEnt ()) != NULL)) {
                            trataError (close (0));
                            trataError (open (fichEnt, O_RDONLY));
                        }
                        if (i > 0) { //el proceso i leerá de pipes[i-1]

                            (*) 

                        }

                        if (i < (numCmds - 1)) { //el proceso i escribirá en pipes[i]
                            ....
                        }
                        ...
                        trataError (execvp (obtenCmd(i), obtenArgs(i)));
                    } //fin de hijo
                } //fin for
            }
        }
    }
}
```

```
...
    trataError (waitpid (pid, &status, 0));
} //fin "si no error sintactico"
} //fin while !finFichero
return(0);
} //fin main
```

**SE PIDE:**

Suponiendo que en la línea de comandos se suministra:

**cmd1 | cmd2 | cmd3**

Supuesto el código anterior. Conteste **RAZONADAMENTE** a las siguientes preguntas:

1. ¿Cuántos pipes se crean? ¿Cuántos procesos crea el padre?
2. ¿Quién crea los pipes, el padre o los hijos? ¿Por qué?
3. ¿Quién realiza las redirecciones, el padre o los hijos? ¿Por qué?
4. Escriba las sentencias necesarias en el recuadro indicado con (\*) para que el proceso i lea del pipe [i-1]