

Unidad 1: Actividad 7

CAPTURAS:

```
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <sys/wait.h>

int main() {
    int fd1[2], fd2[2]; // Pipes para comunicación abuelo-hijo
    int fd3[2], fd4[2]; // Pipes para comunicación hijo-nieto
    pid_t pid_hijo, pid_nieto;

    // Crear pipes
    pipe(fd1); // Pipe fd1 (abuelo -> hijo)
    pipe(fd2); // Pipe fd2 (hijo -> abuelo)
    pipe(fd3); // Pipe fd3 (hijo -> nieto)
    pipe(fd4); // Pipe fd4 (nieto -> hijo)

    // Crear proceso hijo
    pid_hijo = fork();

    switch (pid_hijo) {
        case -1:
            // Error al crear el proceso hijo
            perror("Error al crear el proceso hijo");
            return 1;

        case 0:
            // Proceso HIJO
            pid_nieto = fork(); // Crear proceso nieto

            switch (pid_nieto) {
                case -1:
                    // Error al crear el proceso nieto
                    perror("Error al crear el proceso nieto");
                    return 1;

                case 0:
                    // Proceso NIETO
                    {
                        char mensaje_padre[100];
                        // Recibe mensaje del hijo (padre)
                        read(fd3[0], mensaje_padre, sizeof(mensaje_padre));
                        printf("NIETO recibe mensaje de su padre: %s\n", mensaje_padre);

                        // Enviar mensaje al hijo (padre)
                        char mensaje_nieto[] = "Saludos del Nieto...";
                        write(fd4[1], mensaje_nieto, strlen(mensaje_nieto) + 1);
                        printf("NIETO envia un mensaje a su padre.\n");
                    }
            }
            break;

        default:
            // Proceso HIJO
            {
                char mensaje_abuelo[100];
                // Recibe mensaje del abuelo
                read(fd1[0], mensaje_abuelo, sizeof(mensaje_abuelo));
                printf("HIJO recibe mensaje del abuelo: %s\n", mensaje_abuelo);

                // Enviar mensaje al nieto
                char mensaje_hijo[] = "Saludos del PADRE...";
                write(fd3[1], mensaje_hijo, strlen(mensaje_hijo) + 1);
                printf("HIJO envia mensaje a su hijo.\n");

                // Recibe mensaje del nieto
                char mensaje_nieto[100];
                read(fd4[0], mensaje_nieto, sizeof(mensaje_nieto));
                printf("HIJO recibe mensaje de su hijo (Nieto): %s\n", mensaje_nieto);

                // Enviar mensaje al abuelo
                char mensaje_hijo_abuelo[] = "Saludos del HIJO...";
                write(fd2[1], mensaje_hijo_abuelo, strlen(mensaje_hijo_abuelo) + 1);
                printf("HIJO envia mensaje a su padre (Abuelo)...\n");
            }
    }
}
```

```

        wait(NULL); // Espera al proceso nieto
    }
    break;
}
break;

default:
    // Proceso ABUELO
    {
        // Enviar mensaje al hijo
        char mensaje_abuelo[] = "Saludos del ABUELO...";
        write(fd1[1], mensaje_abuelo, strlen(mensaje_abuelo) + 1);
        printf("ABUELO envía mensaje al hijo...\n");

        // Recibir mensaje del hijo
        char mensaje_hijo[100];
        read(fd2[0], mensaje_hijo, sizeof(mensaje_hijo));
        printf("ABUELO recibe mensaje del hijo: %s\n", mensaje_hijo);

        wait(NULL); // Espera al proceso hijo
    }
    break;
}

```

```

        wait(NULL); // Espera al proceso hijo
    }
    break;
}

return 0;
}

```

```

martin@ldapserver:~$ ./act7
ABUELO envía mensaje al hijo...
HIJO recibe mensaje del abuelo: Saludos del ABUELO...
HIJO envía mensaje a su hijo.
NIETO recibe mensaje de su padre: Saludos del PADRE...
HIJO recibe mensaje de su hijo (Nieto): Saludos del Nieto...
HIJO envía mensaje a su padre (Abuelo)...
NIETO envía un mensaje a su padre.
ABUELO recibe mensaje del hijo: Saludos del HIJO...
martin@ldapserver:~$ █

```