

1º PARCIAL 2023

(T2)

① dode el código que se ejecuta como \$ ./ej22

```
int main (int argc, char **argv) {
```

```
    int i;
```

```
    for (i=0; i<atoi(argv[1]); i++) {
```

```
        fork();
```

```
        printf ("soy %d hijo de %d\n", getpid(), getppid());
```

```
        if (atoi(argv[2]) == 1) {
```

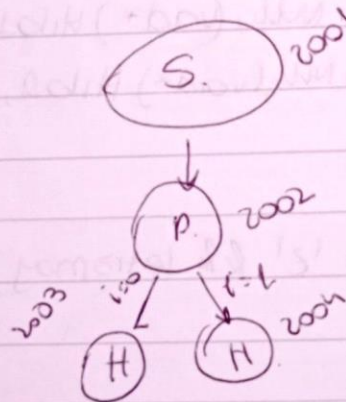
```
            sleep(2);
```

```
        } else { wait(0); }
```

```
    }
```

```
    return 0;
```

```
}
```



Print soy 2002 hijo de 2001

" 2003 " " 2002

" 2004 " " 2002

② No se generan procesos zombies / huérfanos por lo llamado a wait(0).



4) Escribir un proceso que cree 2 hilos graba el abecedario en minúsculas en el archivo abcmín.txt y el otro hilo lo escribe en mayúsculas en abcmay.txt.  
El proceso termina cuando recibe la señal SIGUSR1

```
void sig1(int);  
void Hilo1(void);  
void Hilo2(void);  
char letamín = 'a';  
char letamay = 'A';  
int salir = 0; } globales que los ven los handlers  
int main(void) {  
    signal(SIGUSR1, sig1);  
    FILE *fd1 = fopen("abcmín.txt", "w");  
    FILE *fd2 = fopen("abcmay.txt", "w");  
    pthread_t h1;  
    pthread_t h2;  
    pthread_create(&h1, NULL, (void*) Hilo1, NULL);  
    pthread_create(&h2, NULL, (void*) Hilo2, NULL);  
    while (!salir) {  
        if (letamín != 'z' && letamay != 'z') {  
            pause();  
        }  
        pthread_join(h1, NULL);  
        pthread_join(h2, NULL);  
        fclose(fd1);  
        fclose(fd2);  
        return 0;  
    }
```

```
void H1b1 (void) {
```

```
    while (letromin ≤ 'z') {
```

```
        fprintf (fd1, "%c\n", letromin);
```

```
        letromin++;
```

```
    }
```

```
}
```

```
void H1b2 (void) {
```

```
    while (letromay ≤ 'z') {
```

```
        fprintf (fd2, "%c\n", letromay);
```

```
        letromay++;
```

```
    }
```

```
}
```

```
void h-ag1 (int s) {
```

```
    solir = 1
```

```
}
```



## PARCIAL 2023 (T1)

- 4) Escribir un proceso que al recibir la señal SIGUSR1 crea un hilo. La tarea del hilo es: incrementar en 1 la variable global incrementada en cero. Retornar el valor de la var antes de terminar. El proceso finaliza luego de 1 min de ejecución.

```
void Hilo1 (void);
```

```
void hsig1 (int);
```

```
void halarm (int);
```

```
int varglobal = 0;
```

```
int salir = 0;
```

```
int main () {
```

```
    pthread_t ht;
```

```
    pthread_create(&ht, NULL, hsig1, NULL);
```

```
    signal(SIGUSR1, hsig1);
```

```
    signal(ALRM, halarm);
```

```
    alarm(60);
```

```
    while (!salir) {
```

```
        pause();
```

```
        pthread_join(ht, NULL);
```

```
    }
```

```
    pthread_exit(NULL);
```

```
    return 0;
```

```
}
```

```
void Hilo1 (void) {
```

```
    varglobal ++;
```

```
    printf("/d ln", varglobal);
```

```
}
```

```
void halarm (int s) {
```

```
    salir = 1;
```

```
void hsig1 (int s) {
```

```
    pthread_create(&ht, NULL, (void*)Hilo1,
```

```
    NULL);
```

```
    pthread_join(ht, NULL);
```

```
}
```