

Informe Lab3 Sistemes Operatius

Aran Roig i Biel Palomar

Març 2025



1 Introducció

En aquest document veurem l'implementació dels nostres scripts de la pràctica 3. Que hem fet per controlar errors, i quines comandes hem utilitzat per a fer certes operacions. Totes les comandes utilitzades son dins del tutorial o que s'han fet servir a classe.

2 Implementació i proves

Començant des de la funció main del programa, primer es comproven que el nombre d'arguments donats sigui el correcte i aleshores es guarden respectivament en les variables `lim_alto`, `lim_medio`, `lim_bajo` i `N`

```
if(argc != 5) {
    printf("%s <lim_alto> <lim_medio> <lim_bajo> <N>\n", argv[0]);
    exit(1);
}

float lim_alto = atof(argv[1]);
float lim_medio = atof(argv[2]);
float lim_bajo = atof(argv[3]);
N = atoi(argv[4]); // N està declarada com a variable global de tipus int
```

Aleshores comprovem que `N` no sigui un nombre negatiu i en cas de ser-ho, mostrem un error a l'usuari. Després procedim a crear les dues pipes, que ja havíem inicialitzat com a variables globals fent `pipe(pipe1)` i `pipe(pipe2)`;

Un cop fet això, assignem a 0 els recursos actuals que té cada país, que guardarem a les variables globals `recursosPais1` i `recursosPais2`. Després, el procés pare procedeix a fer dos `fork()`; per crear els processos fills, pero just abans bloquejem les senyals de tipus `SIGUSR1` per així fer que quedin bloquejades en els processos fills. Cada procés fill seguirà el fil d'execució de la funció `fill` i un cop creat els fills el procés principal desbloquejarà la senyal `SIGUSR1`

```
sigemptyset(&mask);
sigaddset(&mask, SIGUSR1);
sigprocmask(SIG_BLOCK, &mask, &oldMask);
int child_pid1 = fork(), child_pid2 = fork(); // Es creen els fills
if(child_pid1 == 0){
    fill();
    return 1;
}
if(child_pid2 == 0){
    fill();
    return 1;
}
sigprocmask(SIG_UNBLOCK, &mask, &oldMask);
```

2.1 Processos fills

Els processos fills, executen la mateixa funció. Aquesta es distribueix en diferents parts: Primer de tot, creem la seed del número random, fent servir en comptes del time el número del pid, i afegim una funció (buida) per tractar les senyals `SIGUSR1`

```
srand(getpid());
signal(SIGUSR1, sigusr1);
```

Seguidament fem un bucle que iteri tants cops com anys simularem.

Per cada simulació, primer de tot, si tenim alguna senyal SIGUSR1 per processar, el procés segueix amb la seva execució. En cas contrari, esperem a rebre-la. Tot això ens ho fa la funció `sigsuspend(&oldMask)`; degut a que SIGUSR1 està bloquejada. Un cop processada la senyal, llegim el límit que ens ha passat el pare. Generem un número aleatori en mòdul del límit, així sempre estarà per sota, i enviem la sol·licitud al pare.

```
for(int i = 0; i < N; i++){
    sigsuspend(&oldMask);

    float limitAnual;
    read(pipe1[0], &limitAnual, sizeof(float)); // Lee mensaje del padre

    printf("Limit anual es %f\n", limitAnual);

    float recursosExtreure = rand() % ((int) limitAnual);
    write(pipe2[1], &recursosExtreure, sizeof(float));
}
```

Un cop fet el bucle tanquem les tuberries i s'ha acabat el procés fill.

2.2 Procés Pare

Per al procés del pare haurem de fer un bucle com el dels fills, que es faci tants cops com anys simularem. Dins del bucle el primer que farem és calcular el límit anual fent servir la taula proporcionada a l'enunciat de la pràctica.

```
printf("* AÑO %d \n", i);
printf("[Coordinador] Los recursos disponibles para el año son %f\n", recursos );
//Calculo limites
float limiteAnual = 0;
if(recursos <= 1000 && recursos > 750){
    limiteAnual = lim_alto;
}
else if(recursos < 750 && recursos > 450){
    limiteAnual = lim_medio;
}
else if(recursos <= 450 && recursos > 0){
    limiteAnual = lim_bajo;
}

float llegitsPais1, llegitsPais2;
printf("[Coordinador] El límite de extracción para el año en curso es %f\n", limiteAnual);
```

Seguidament escriurem aquest límit a la pipe1 (fill1) i llegirem el valor que ens retorna com a sol·licitud aquest fill.

```
write(pipe1[1], &limiteAnual, sizeof(float));
kill(child_pid1, SIGUSR1);
```

```
read(pipe2[0], &llegitsPais1, sizeof(float));
printf("[Coordinador] El pais 1 solicita extraer %f\n", llegitsPais1);
```

Un cop hem rebut la sol·licitud del fill fem la resta als recursos actuals i ho sumem al sumatori dels recursos demanats per aquest fill.

```
printf("[Coordinador] La solicitud del pais 1 se ha aprobado\n");
recursos = recursos - llegitsPais1;
if(recursos < 0) recursos = 0;
recursosPais1 = recursosPais1 + llegitsPais1;
```

Ara repetim el mateix codi per al segon fill i just abans d'acabar el codi del bucle executem la funció que calcula els recursos per a l'any vinent.

2.3 Funció recursos

La funció dels recursos ens permet calcular quants recursos tindrem el pròxim any de simulació. Per a fer-ho primer de tot calculem el P_b seguint la taula de l'enunciat

```
float pb = 0.05;
if(recursosActuals <= 750 && recursosActuals > 450){
    pb = 0.20;
}
```

Després calculem el P_v seguint la taula i aplicant un aspecte random a la selecció.

```
float pv = 0;
srand(time(NULL));
int cas = rand() % 100; // num de 0 a 99

if (cas < 75) {
    printf("[Evento] Este año se han producido condiciones normales de recuperación \n");
    pv = 0;
} else if (cas < 90) {
    printf("[Evento] Este año se han producido condiciones adversas de recuperación \n");
    pv = -0.15;
} else {
    printf("[Evento] Este año se han producido condiciones favorables de recuperación \n");
    pv = 0.1;
}
```

Llavors sumem els dos percentatges per a obtenir el total. Si el total és negatiu, la suma serà 0.

```
float prob = pb + pv;
if(prob < 0){
    prob = 0;
}
```

Finalment, calculem l'increment de recursos i ho sumem als recursos actuals.

```
printf("[Evento] El porcentaje de recuperación es del %f\n", prob);
float increment = recursosActuals * prob;
if(increment + recursosActuals > 1000){
    recursos = 1000;
```

```
    }else{  
        recursos = recursosActuals + increment;  
    }  
    printf("[Coordinador] Los recursos se recuperaron en %f unidades\n", increment);
```

Amb tot això fet ja hem acabat el nostre programa.

3 Conclusions i valoracions personals

En conclusió, aquesta pràctica ens ha permès practicar la comunicació entre processos i com gestionar els fills amb diverses pipes bidireccional.

Personalment, opinem que la pràctica està ben pensada i ajuda a assolir els coneixements explicats a classe.

4 Contribució

Aran Roig Serra 50%

Biel Palomar González 50%