Roman Atanasiu u213937
Xavier Cañadas u213934

1. Explain why two pipes are enough even if there are several children

Una pipe és un mètode de comunicació entre 2 processos, però que no està lligat a 2 processos en concret. A més, no hi ha comunicació entre fills, per tant, no cal una altra pipe per comunicar-se entre processos fills. A més, tota la informació queda emmagatzemada en la pipeA al fins que un fill la recull, i després del wait, tota la informació está en la pipeB, per tant no calen més pipes.

2. Explain how the ending is handled (when the standard input of the father closes)
Tanquem l'extrem de lectura de la pipeB, ja que és l'últim extrem que queda per tencar i fem un exit code 0 ja que el programa ha acabat d'executar-se sense cap error.

```
    printf("FINISHED\n");
    while(wait(NULL) == -1);

    // Now that is finished, write all the results
    Result res;

    while((nBytesRead = read(pipeB[0], &res, sizeof(res)) ) > 0) {
        printf("The CRC of block #%d is %u \n", res.nBlock, res.crc);
    }
    close(pipeB[0]);
    exit(0);
}
```

El close dels fitxers els fan els fills abans de morir, això és degut a que els fills obren els fitxers, i el pare no utilitza fitxers. I les altres pipes o bé s'han tancat abans d'executar el procés ja que no és fan servir, o bé després que ja no s'utilitza en un procés. Per tant, no cal tancar res més després que el programa fagi l'execució.

```
    }

}
close(pipeA[0]);
close(pipeB[1]);
close(fd);
close(fdCRC);

exit(0);
```

3. An alternative of using file locks would be to use a named semaphore to make any access to the file
exclusive. Which disadvantages has this solution?

Els files locks i unlocks es poden utilitzar per tancar certa part del fitxer, per tant, si un procés entra en una part no lockejada del file podrà seguir funcionant, mentres que amb un semàfor s'hauria d'esperar fins que un altre procés acabi la seva execució i comenci la seva.