

# COMPROC2 - Grup 62 - 22-23Q1

## uBash

Crea un programa, que li direm ubash, modificant el fitxer ubash.c que es dona a l'enunciat. Aquest serà el nostre mini intèrpret de comandes, que haurà de suportar pipes sense nom.

En general, t'hauràs de centrar en la funció `void processa_comanda(int num_arguments, char *arguments[])`. Aquesta funció es crida cada cop que un usuari fa enter al nostre intèrpret de comandes. Prova la versió donada de `ubash.c`, que només repeteix els arguments que li arriben, per assegurar-te que entens com funciona.

Per exemple:

```
$ ./ubash
ubash> grep main ubash.c
grep main ubash.c
ubash> [Ctrl^D]
```

Com veus, pots sortir de ubash enviant un EOF (Ctrl + D) al programa.

L'exercici consisteix en fer el següent:

1. Fes que quan ubash rebi una línia d'input, faci fork del procés, i muti a la comanda que hagi escrit l'usuari. ubash ha d'esperar a que acabi la comanda abans de demanar-ne una altra, i ha d'imprimir per pantalla el valor de retorn d'aquesta, un cop acabi.
2. Ara, introdueix suport per a pipes sense nom, com bash. Si l'usuari posa 2 o més comandes, amb el caràcter de pipe "|" entre elles, ubash haurà de crear tants processors com comandes, i tantes pipes sense nom com caràcters de pipe hagi posat l'usuari, i connecti els processos creats entre ells com faria la consola. En aquest cas, volem imprimir el valor de retorn de l'última comanda que ha posat l'usuari.
3. [Si us sobra temps] Afegeix suport per redirigir fitxers d'entrada a les comandes (comanda < fitxer)

Afegeix també **un makefile** per a compilar ubash, i **comprova el valor de retorn de totes les crides a sistema**. Aneu guardant les versions d'ubash que us funcionin (després de fer el punt 1 i 2).

Entregueu **un sol fitxer .tar.gz** que contingui el Makefile, i totes les versions d'ubash que hagueu fet (inclús la que tingueu a mitges al acabar, si és el cas).

Exemple de l'apartat 1:

```
$ ./ubash
ubash> grep main ubash.c
int main(int argc, char *argv[])
Comanda retorna 0
ubash> date
diumenge, 11 de desembre de 2022, 20:21:08 CET
Comanda retorna 0
ubash> so mola
No s'ha trobat l'ordre (exec ha fallat)
Comanda retorna 1
ubash> [Ctrl^D]
```

Exemple de l'apartat 2:

```
$ ./ubash
ubash> grep main ubash.c | wc -c
33
Comanda retorna 0
ubash> date | wc -c
46
Comanda retorna 0
ubash> head -n 1 ubash.c | cat | cat | cat | cat | cat
#define _GNU_SOURCE
Comanda retorna 0
ubash> [Ctrl^D]
```

Pistes:

- `exec lp` no et servirà per mutar el programa, perquè no saps quants arguments té. Però potser algun dels seus cosins sí (`man 3 exec`).
- Recorda que el primer argument que rep un programa a `argv` és el mateix nom del programa.
- Podeu assumir que mai hi haurà més de 8 pipes en una sola línia de comandes.
- Podeu assumir també que els inputs són correctes sintàcticament (no hi ha pipes al principi o al final, per exemple)