INF02JIN - getopt

30.IV.2020

argc / argv <getopt.h>

① struct option
"gravity", required_argment, 0, 'g'

② getopt_long → int option courte

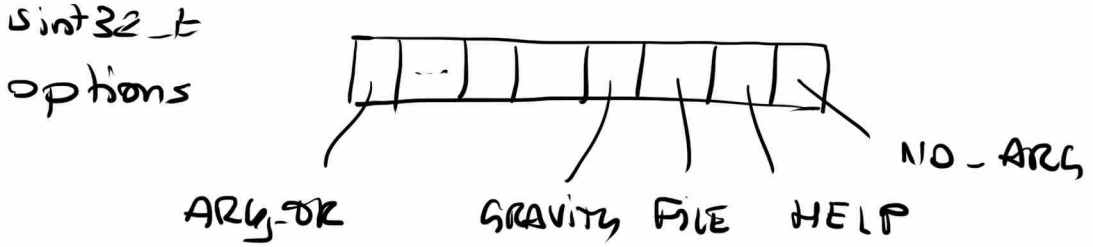
③ optarg → chaine de caracteres

./app --gravity = 9.81

"9.81" → optarg
variable globale
définie dans getopt

./app --file = "data.txt"

↳ optarg



Gestion des arguments --file = ---
--gravity = ---

① Créer une structure qui va contenir toutes les arguments des options. sOptArg

② Modifier le prototype de processArg pour accepter une variable de type sOptArg modifiable

③ Mettre à jour dans processArg les champs de la variable de type sOptArg en fonction des options

15:22

Structure sOptArg :

```
char filename[128];   // -- file
double gravity;       // --gravity
```

Prototype processArg

```
(int argc, char *argv[], sOptArg *opt);
```

Gestion des Arguments

--file= "data.test" copier dans filename
--gravity= "9.81" l'interpréter comme un réel → gravity.

Main

1) créer une variable pour accueillir les arguments des options
2) passer cette variable à processArg
3) au retour, afficher les arguments mis à jour

15:45

---

main

opt . filename
. gravity

ars

1) NO_ARG → usage()
   ret = 254.
2) HELP → usage()
   ret = 0
3) BAD_ARG → ret = 253
   (+ msg d'erreur)

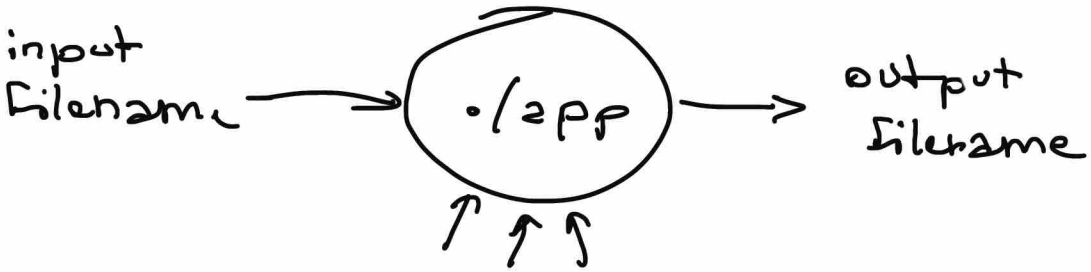
4)
ARG_OK et FILE
exploiter opt.filename
ret = 0
ARG_OK et GRAVITY
exploite opt.gravity

16h10

Préparation Labo#6 "Traitement d'image"

- ./app [--options]

  --ifile= input filename
  --ofile= output filename
  --help



  --info
  --black
  --togray
  --negate
  --hmirror
  --vmirror
  --rotate = angle (° réel)
  --pixelisation
  --custom

.h { eArgCode, sOptArg }

.c [processArg, main traiter la variable arg.]

?