

# Examen de Système (FLIN502)

Michel Meynard

Durée : 2 heures

Tous documents autorisés

14 décembre 2009

## 1 Analyse d'un programme

Soit le programme `prog.c` suivant :

```
#include <stdio.h>
#include <stdlib.h> /* atoi */
#include <unistd.h> /* sprintf */

int main(int argc, char *argv[], char* env[]){
    if(argc<2){
        fprintf(stderr,"Erreur de syntaxe ! %s n1 n2 n3 ...\n",argv[0]);
        return 1;
    }else if (argc==2){
        printf("Résultat : %s\n",argv[1]);
        return 0;
    }else{
        char s[100];
        sprintf(s,"%d",atoi(argv[1])+atoi(argv[2]));
        char *param[argc];
        param[0]=argv[0];
        param[1]=s;
        for(int i=2;i<argc-1;i++)
            param[i]=argv[i+1];
        param[argc-1]=NULL;
        execvp(argv[0],param);
    }
}
```

### 1.1 Questions

Répondre à ces questions en **2 à 3 phrases** claires et correctement écrites (orthographe et grammaire).

1. Quelle est la signature (déclaration) de la fonction `atoi` et que réalise-t-elle ?
2. Quelle est la signature (déclaration) de la fonction `sprintf` et que réalise l'appel suivant : `sprintf(s,"%d",3+4)` ?
3. La fonction `execvp` est-elle un appel système ou une fonction de bibliothèque ? Quel est son rôle ?
4. Que se passe-t-il si on appelle ce programme avec les arguments suivants : `prog toto 1 titi 4` ?
5. Décrire l'objectif réalisé par ce programme et donnez-lui un nom plus parlant que `prog.c`.

## 2 Fichier csv

Un fichier csv (character separated value), est un fichier texte constitué d'articles séparés par un retour ligne (`'\n'`) et où chaque article est composé de champs séparés par un caractère délimiteur, par exemple `':'`. Le fichier suivant `rep.csv` est un exemple de fichier csv implémentant un répertoire téléphonique où chaque article est de la forme `nom:numéro:type-téléphone`.

```
michel:0412345678:domicile
lea:0612345678:mobile
paul:0142356658:bureau
michel:0645678912:mobile
...
```

Les articles ne sont pas triés et ce genre de fichier est **non borné** (il est donc impossible de le recopier en mémoire centrale dans un tableau) !

## 2.1 Questions

1. Quelle fonction utiliser pour lire un article du fichier ?
2. Connaissez-vous une fonction de bibliothèque permettant d'extraire un champ d'un article ?
3. Ecrire l'algorithme d'une fonction `char* getChamp(char *s, int i, char delim)` permettant de retourner la chaîne de caractères correspondant au ième champ de `s`. Par exemple :  
`getChamp("michel:0645678912:mobile",1,':')` retourne "0645678912". Les champs sont numérotés de 0 à `n-1` et sont séparés par le caractère `delim`.
4. Ecrire la fonction C `getChamp`.
5. On veut écrire un programme `tel.c` permettant d'afficher tous les numéros de téléphone d'une personne du répertoire `rep.csv`. Par exemple :

```
>tel rep.csv michel
domicile      0412345678
mobile        0645678912
```

Ecrire l'algorithme de ce programme.

6. Ecrire le programme `tel.c`

## 3 Parallélisme

On souhaite paralléliser le programme `tel` de la façon suivante : le processus parent crée autant d'enfants que de Kilo-octets du fichier `rep.csv` et chaque enfant ne traite que les articles commençant dans son propre Kilo-octet.

### 3.1 Questions

1. Comment le parent peut-il connaître le nombre d'enfants à générer ?
2. Le dernier enfant peut-il ne rien faire ?
3. Comment un enfant peut-il se rendre au début de son propre Ko ?
4. Le fichier doit-il être ouvert une seule ou plusieurs fois ? Décrivez les avantages et inconvénients.
5. Ecrire l'algorithme de cette version parallèle.