Coding Styles

Ensimag 3A SLE

Matthieu Moy

Matthieu.Moy@imag.fr

2016-2017

Sommaire

Une page de journal ...

europe

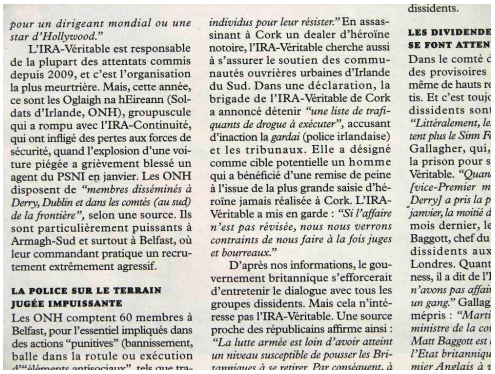
IRLANDE DU NORD

Ils résisteront toujours aux Britanniques

ROYAUME-UNI

Nick Clegg ou le troisième homme

Regardons d'un peu plus près ...



pour un dirigeant mondial ou une star d'Hollywood."

L'IRA-Véritable est responsable de la plupart des attentats commis depuis 2009, et c'est l'organisation la plus meurtrière. Mais, cette année, ce sont les Oglaigh na hEireann (Soldats d'Irlande, ONH), groupuscule qui a rompu avec l'IRA-Continuité, qui ont infligé des pertes aux forces de sécurité, quand l'explosion d'une voiture piégée a grièvement blessé un agent du PSNI en janvier. Les ONH disposent de *"membres disséminés à Derry, Dublin et dans les comtés (au sud) de la frontière"*, selon une source. Ils sont particulièrement puissants à Armagh-Sud et surtout à Belfast, où leur commandant pratique un recrutement extrêmement agressif.

LA POLICE SUR LE TERRAIN JUGÉE IMPUISSANTE

Les ONH comptent 60 membres à Belfast, pour l'essentiel impliqués dans des actions "punitives" (bannissement, balle dans la rotule ou exécution

individus pour leur résister." En assassinant à Cork un dealer d'héroïne notoire, l'IRA-Véritable cherche aussi à s'assurer le soutien des communautés ouvrières urbaines d'Irlande du Sud. Dans une déclaration, la brigade de l'IRA-Véritable de Cork a annoncé détenir *"une liste de trafiquants de drogue à exécuter"*, accusant d'inaction la *gardai* (police irlandaise) et les tribunaux. Elle a désigné comme cible potentielle un homme qui a bénéficié d'une remise de peine à l'issue de la plus grande saisie d'héroïne jamais réalisée à Cork. L'IRA-Véritable a mis en garde : *"Si l'affaire n'est pas révisée, nous nous verrons contraints de nous faire à la fois juges et bourreaux."*

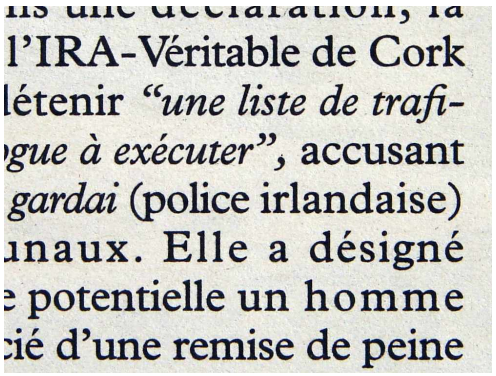
D'après nos informations, le gouvernement britannique s'efforcerait d'entretenir le dialogue avec tous les groupes dissidents. Mais cela n'intéresse pas l'IRA-Véritable. Une source proche des républicains affirme ainsi : *"La lutte armée est loin d'avoir atteint un niveau susceptible de pousser les Bri-*

dissidents.

LES DIVIDENDE SE FONT ATTEN

Dans le comté d des provisoires même de hauts r tis. Et c'est touj dissidents sont *"Littéralement, la tent plus le Sinn F* Gallagher, qui, la prison pour s Véritable. *"Quan [vice-Premier m Derry] a pris la p* janvier, la moitié d mois dernier, le Baggott, chef du dissidents aux Londres. Quant ness, il a dit de l' *n'avons pas affai un gang."* Gallag mépris : *"Marti ministre de la co Matt Baggott est l'Etat britanniqu mier Anglais à*

Et encore un peu plus près ...



...ils une déclaration, la
l'IRA-Véritable de Cork
létenir *"une liste de trafi-*
gue à exécuter", accusant
gardai (police irlandaise)
...naux. Elle a désigné
...e potentielle un homme
...cié d'une remise de peine

Une page de journal ...

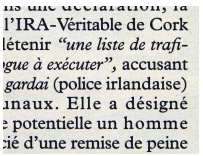
- Page découpée en plusieurs articles,
- Chaque article a un titre,
- Les articles longs sont découpés avec des sous-titres.

Regardons d'un peu plus près ...

- Disposition en colonnes
  (≈ 40 caractères de large)
  ⇒ lecture plus facile (retour des yeux à la ligne suivante).
- Découpage du texte en paragraphe
  ⇒ Saut de paragraphe = petite pause, passage à une autre notion.

Et encore un peu plus près ...

- Placement des espaces et ponctuation
  ⇒ lecture fluide.
  - Pas d'espaces à l'intérieur des parenthèses
  - Espaces autour des parenthèses
  - Espace après une virgule mais pas avant
  - ...



...ils une déclaration, la
l'IRA-Véritable de Cork
létenir *"une liste de trafi-*
gue à exécuter", accusant
gardai (police irlandaise)
...naux. Elle a désigné
...e potentielle un homme
...cié d'une remise de peine

Et si on faisait autrement ?

Tout informaticien doit connaître le langage C.C ' est une espèce d ' espéranto de l'informatique .Les autres langages fournissent en effet souvent une interface avec C(ce qui leur permet en particulier de s'interfacer plus facilement avec le système d ' exploitation) ou sont eux-mêmes écrits en C.D'autre part c ' est le langage de base pour programmer les couches basses des systèmes informatiques. Par exemple ,on écrit rarement un pilote de périphérique en Ada ou Java. Finalement ,en compilation, C est souvent choisi comme cible de langages de plus haut niveau. Toutefois, il est peu probable(ou plus exactement ,peu souhaitable) qu'un ingénieur informaticien soit confronté à de gros développements logiciels entièrement en C.L'objectif pédagogique du projet est donc surtout de montrer comment C peut servir d ' interface entre les langages de haut niveau et les couches basses de la machine.Plus précisément, les objectifs du stage C de première année sont : Apprentissage de C(en soi ,et pour la démarche qui consiste à apprendre un nouveau langage). Lien du logiciel avec les couches basses de l'informatique ,ici logiciel de base et architecture. Le premier projet logiciel un peu conséquent ,à développer dans les règles de l ' art(mise en œuvre de tests ,documentation, démonstration du logiciel, partage du travail, ...)Lien avec les autres modules de première année(théorie des langages) et de deuxième année (compilation, système)

Tentons de faire mieux ...

Tout informaticien doit connaître le langage C. C'est une espèce d'espéranto de l'informatique. Les autres langages fournissent en effet souvent une interface avec C (ce qui leur permet en particulier de s'interfacer plus facilement avec le système d'exploitation) ou sont eux-mêmes écrits en C. D'autre part c'est le langage de base pour programmer les couches basses des systèmes informatiques. Par exemple, on écrit rarement un pilote de périphérique en Ada ou Java. Finalement, en compilation, C est souvent choisi comme cible de langages de plus haut niveau. Toutefois, il est peu probable (ou plus exactement, peu souhaitable) qu'un ingénieur informaticien soit confronté à de gros développements logiciels entièrement en C. L'objectif pédagogique du projet est donc surtout de montrer comment C peut servir d'interface entre les langages de haut niveau et les couches basses de la machine. Plus précisément, les objectifs du stage C de première année sont : Apprentissage de C (en soi, et pour la démarche qui consiste à apprendre un nouveau langage). Lien du logiciel avec les couches basses de l'informatique, ici logiciel de base et architecture. Le premier projet logiciel un peu conséquent, à développer dans les règles de l'art (mise en œuvre de tests, documentation, démonstration du logiciel, partage du travail, ...) Lien avec les autres modules de première année (théorie des langages) et de deuxième année (compilation, système)

Tentons de faire mieux ...

Tout informaticien doit connaître le langage C. C'est une espèce d'espéranto de l'informatique. Les autres langages fournissent en effet souvent une interface avec C (ce qui leur permet en particulier de s'interfacer plus facilement avec le système d'exploitation) ou sont eux-mêmes écrits en C. D'autre part c'est le langage de base pour programmer les couches basses des systèmes informatiques. Par exemple, on écrit rarement un pilote de périphérique en Ada ou Java. Finalement, en compilation, C est souvent choisi comme cible de langages de plus haut niveau.
Toutefois, il est peu probable (ou plus exactement, peu souhaitable) qu'un ingénieur informaticien soit confronté à de gros développements logiciels entièrement en C. L'objectif pédagogique du projet est donc surtout de montrer comment C peut servir d'interface entre les langages de haut niveau et les couches basses de la machine. Plus précisément, les objectifs du stage C de première année sont :
Apprentissage de C (en soi, et pour la démarche qui consiste à apprendre un nouveau langage).
Lien du logiciel avec les couches basses de l'informatique, ici logiciel de base et architecture.
Le premier projet logiciel un peu conséquent, à développer dans les règles de l'art (mise en œuvre de tests, documentation, démonstration du logiciel, partage du travail, ...)
Lien avec les autres modules de première année (théorie des langages) et de deuxième année (compilation, système)

Tentons de faire mieux ...

Tout informaticien doit connaître le langage C. C'est une espèce d'espéranto de l'informatique. Les autres langages fournissent en effet souvent une interface avec C (ce qui leur permet en particulier de s'interfacer plus facilement avec le système d'exploitation) ou sont eux-mêmes écrits en C. D'autre part c'est le langage de base pour programmer les couches basses des systèmes informatiques. Par exemple, on écrit rarement un pilote de périphérique en Ada ou Java. Finalement, en compilation, C est souvent choisi comme cible de langages de plus haut niveau.

Toutefois, il est peu probable (ou plus exactement, peu souhaitable) qu'un ingénieur informaticien soit confronté à de gros développements logiciels entièrement en C. L'objectif pédagogique du projet est donc surtout de montrer comment C peut servir d'interface entre les langages de haut niveau et les couches basses de la machine. Plus précisément, les objectifs du stage C de première année sont :

- Apprentissage de C (en soi, et pour la démarche qui consiste à apprendre un nouveau langage).
- Lien du logiciel avec les couches basses de l'informatique, ici logiciel de base et architecture.
- Le premier projet logiciel un peu conséquent, à développer dans les règles de l'art (mise en œuvre de tests, documentation, démonstration du logiciel, partage du travail, ...)
- Lien avec les autres modules de première année (théorie des langages) et de deuxième année (compilation, système)

Tentons de faire mieux ...

Tout informaticien doit connaître le langage C. C'est une espèce d'espéranto de l'informatique. Les autres langages fournissent en effet souvent une interface avec C (ce qui leur permet en particulier de s'interfacer plus facilement avec le système d'exploitation) ou sont eux-mêmes écrits en C. D'autre part c'est le langage de base pour programmer les couches basses des systèmes informatiques. Par exemple, on écrit rarement un pilote de périphérique en Ada ou Java. Finalement, en compilation, C est souvent choisi comme cible de langages de plus haut niveau.

Toutefois, il est peu probable (ou plus exactement, peu souhaitable) qu'un ingénieur informaticien soit confronté à de gros développements logiciels entièrement en C. L'objectif pédagogique du projet est donc surtout de montrer comment C peut servir d'interface entre les langages de haut niveau et les couches basses de la machine. Plus précisément, les objectifs du stage C de première année sont :

- Apprentissage de C (en soi, et pour la démarche qui consiste à apprendre un nouveau langage).
- Lien du logiciel avec les couches basses de l'informatique, ici logiciel de base et architecture.
- Le premier projet logiciel un peu conséquent, à développer dans les règles de l'art (mise en œuvre de tests, documentation, démonstration du logiciel, partage du travail, ...)
- Lien avec les autres modules de première année (théorie des langages) et de deuxième année (compilation, système)

Conclusion

- Le fond est important
- ... mais on a du mal à se concentrer sur le fond si la forme est imparfaite !

⇝ C'est pareil pour du code !

Principe de la « fenêtre brisée »

- Une fenêtre brisée

⇒ le vent s'engouffre, des saletés entrent

⇒ « mon voisin n'entretient pas ses fenêtres, pourquoi je le ferais ? »

- Une ligne mal indentée

⇒ « je ne vais pas corriger, ça va faire des conflits dans Git »

⇒ « pourquoi je m'embêterait à formatter mon code proprement ? »

⇒ « pourquoi j'écrirais de la doc, de toutes façons, le code est sale »

⇒ « la dernière fois que j'ai codé, je n'ai pas écrit de tests, ça s'est bien passé quand même ... »

⚠ Attention à la dette technique

Peut-on faire mieux ?

```
struct noeud_t{
 unsigned  val;
  struct noeud_t *fg ;
 struct  noeud_t * fd;
};

int rechercher(unsigned v,struct noeud_t *arbre)
{
  int res;
  if(arbre != NULL) {
   if (v==arbre->val){
    return 1;
   } else if(v<arbre->val){
     return rechercher(v,arbre->fg);
   }
   else
   {
       res =rechercher( v, arbre -> fd ) ;
       return res ;
     }
  } else {
         return 0 ;
  }
}
```

Peut-on faire mieux ?

```
struct noeud_t{
        unsigned  val;
        struct noeud_t *fg ;
        struct  noeud_t * fd;
};

int rechercher(unsigned v,struct noeud_t *arbre)
{
        int res;
        if(arbre != NULL) {
                if (v==arbre->val){
                        return 1;
                } else if(v<arbre->val){
                        return rechercher(v,arbre->fg);
                }else{
                        res =rechercher( v, arbre -> fd ) ;
                        return res ;
                }
        } else {
                return 0 ;
        }
}
```


Peut-on faire mieux ?

```
struct noeud_t {
        unsigned val;
        struct noeud_t *fg;
        struct noeud_t *fd;
};

int rechercher(unsigned v, struct noeud_t *arbre)
{
        int res;
        if (arbre != NULL) {
                if (v == arbre->val) {
                        return 1;
                } else if (v < arbre->val) {
                        return rechercher(v, arbre->fg);
                } else {
                        res = rechercher(v, arbre->fd);
                        return res ;
                }
        } else {
                return 0;
        }
}
```


Peut-on faire mieux ?

```
struct noeud_t {
        unsigned val;
        struct noeud_t *fg;
        struct noeud_t *fd;
};

int rechercher(unsigned v, struct noeud_t *arbre)
{
        int res;
        if (arbre == NULL) {
                return 0;
        } else {
                if (v == arbre->val) {
                        return 1;
                } else if (v < arbre->val) {
                        return rechercher(v, arbre->fg);
                } else {
                        res = rechercher(v, arbre->fd);
                        return res;
                }
        }
}
```


Peut-on faire mieux ?

```
struct noeud_t {
        unsigned val;
        struct noeud_t *fg;
        struct noeud_t *fd;
};

int rechercher(unsigned v, struct noeud_t *arbre)
{
        if (arbre == NULL) {
                return 0;
        } else if (v == arbre->val) {
                return 1;
        } else if (v < arbre->val) {
                return rechercher(v, arbre->fg);
        } else {
                return rechercher(v, arbre->fd);
        }
}
```

⇝ on va s'arrêter là !

Peut-on faire mieux ?

```
struct noeud_t {
        unsigned val;
        struct noeud_t *fg;
        struct noeud_t *fd;
};

int rechercher(unsigned v, struct noeud_t *arbre)
{
        if (arbre == NULL) {
                return 0;
        } else if (v == arbre->val) {
                return 1;
        } else if (v < arbre->val) {
                return rechercher(v, arbre->fg);
        } else {
                return rechercher(v, arbre->fd);
        }
}
```


Directives de codage du noyau Linux

- Utilisées par Linux et beaucoup d'autres projets
- Dispo un peu partout sur le Web (`http://lxr.linux.no/#linux/Documentation/CodingStyle`)
- Bon ou mauvais style : question subjective, mais l'important est d'être homogènes !

"First off, I'd suggest printing out a copy of the GNU coding standards, and NOT read it. Burn them, it's a great symbolic gesture."

Indentation, présentation du code

1. Indentation à 8 caractères : blocs if/while/for/... clairement visibles.
2. Pas de ligne de plus de 80 caractère : plus lisible, affichable sur une petite fenêtre.
3. Une fonction doit tenir sur un écran (en théorie, 1 écran = 25 lignes). Faire des fonctions courtes, qui font une chose et qui le font bien.
4. Si on a l'impression qu'il y a conflit entre les deux premiers points, c'est qu'il faut mieux découper (cf. point 3).
5. Utiliser les lignes vides pour marquer une pause (cf. notion de paragraphe en français)

Nommage des fonctions, variables

- Noms courts et expressifs
  - `ThisVariableIsATemporaryCounter` ⇒ trop long
  - `a`, `var1` ⇒ pas expressifs
- Nom expressif indispensable pour les variables/fonctions globales.

Commentaires

- Expliquez pourquoi votre code est comme il est, et non comment.
- Si le code a besoin de beaucoup de commentaire pour expliquer comment il fonctionne, c'est qu'il est trop complexe ⇒ simplifiez-le plutôt que de le commenter !

Pour nous ...

- Style imposé (lisez l'énoncé des TPs), à respecter strictement
- Facile à respecter si on est bien outillé
- Oui, je vous enlèverai des points pour les espaces en fin de ligne !