

Ordonnanceur

Généré par Doxygen 1.8.17



<b>1 Index des classes</b>	<b>1</b>
1.1 Liste des classes	1
<b>2 Index des fichiers</b>	<b>3</b>
2.1 Liste des fichiers	3
<b>3 Documentation des classes</b>	<b>5</b>
3.1 Référence de la structure defList	5
3.1.1 Documentation des données membres	5
3.1.1.1 next	5
3.1.1.2 processus	5
3.2 Référence de la structure defProcess	6
3.2.1 Documentation des données membres	6
3.2.1.1 execution_time	6
3.2.1.2 name	6
3.2.1.3 priority	6
3.3 Référence de la structure List	6
3.3.1 Description détaillée	6
3.4 Référence de la structure Process	6
3.4.1 Description détaillée	6
<b>4 Documentation des fichiers</b>	<b>7</b>
4.1 Référence du fichier src/defines.h	7
4.2 Référence du fichier src/functions.c	7
4.2.1 Description détaillée	8
4.2.2 Documentation des fonctions	8
4.2.2.1 AffListe()	8
4.2.2.2 ajout_activite()	8
4.2.2.3 createNode()	9
4.2.2.4 registerProcess()	9
4.2.2.5 run()	10
4.2.2.6 step()	10
4.3 Référence du fichier src/functions.h	11
4.3.1 Description détaillée	11
4.3.2 Documentation des fonctions	11
4.3.2.1 AffListe()	11
4.3.2.2 ajout_activite()	12
4.3.2.3 createNode()	12
4.3.2.4 registerProcess()	13
4.3.2.5 run()	13
4.3.2.6 step()	14
4.4 Référence du fichier src/main.c	14
4.4.1 Description détaillée	14

4.4.2 Documentation des fonctions . . . . .	15
4.4.2.1 main() . . . . .	15
4.5 Référence du fichier src/structures.h . . . . .	15
4.5.1 Description détaillée . . . . .	16
<b>Index</b>	<b>17</b>

# Chapitre 1

## Index des classes

### 1.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

<a href="#">defList</a>	.....	5
<a href="#">defProcess</a>	.....	6
<a href="#">List</a>		
	Structure d'une liste chaîné de processus	6
<a href="#">Process</a>		
	Structure d'un processus	6



## Chapitre 2

# Index des fichiers

### 2.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

src/ <a href="#">defines.h</a>		
Fichier de définitions	.....	7
src/ <a href="#">functions.c</a>		
Fichier des fontions du programme	.....	7
src/ <a href="#">functions.h</a>		
Fichier d'en tête des fonctions de fonctions.c	.....	11
src/ <a href="#">main.c</a>		
Main program	.....	14
src/ <a href="#">structures.h</a>		
Fichier de structures	.....	15





## Chapitre 3

# Documentation des classes

### 3.1 Référence de la structure defList

Graphe de collaboration de defList:

#### Attributs publics

- [Process processus](#)
- struct [defList](#) \* [next](#)

#### 3.1.1 Documentation des données membres

##### 3.1.1.1 next

```
struct defList* defList::next
```

Pointeur sur le noeud suivant de la liste chaîné

##### 3.1.1.2 processus

```
Process defList::processus
```

Elément de la liste chaînée de type [Process](#)

La documentation de cette structure a été générée à partir du fichier suivant :

- src/[structures.h](#)

## 3.2 Référence de la structure defProcess

### Attributs publics

- char \* [name](#)
- int [execution\\_time](#)
- int [priority](#)

### 3.2.1 Documentation des données membres

#### 3.2.1.1 execution\_time

```
int defProcess::execution_time
```

Temps du processus

#### 3.2.1.2 name

```
char* defProcess::name
```

Nom du processus

#### 3.2.1.3 priority

```
int defProcess::priority
```

Priorité du processus

La documentation de cette structure a été générée à partir du fichier suivant :

— [src/structures.h](#)

## 3.3 Référence de la structure List

Structure d'une liste chaîné de processus.

```
#include <structures.h>
```

### 3.3.1 Description détaillée

Structure d'une liste chaîné de processus.

La documentation de cette structure a été générée à partir du fichier suivant :

— [src/structures.h](#)

## 3.4 Référence de la structure Process

Structure d'un processus.

```
#include <structures.h>
```

### 3.4.1 Description détaillée

Structure d'un processus.

La documentation de cette structure a été générée à partir du fichier suivant :

— [src/structures.h](#)

## Chapitre 4

# Documentation des fichiers

### 4.1 Référence du fichier src/defines.h

Fichier de définitions.

Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

### 4.2 Référence du fichier src/functions.c

fichier des fonctions du programme

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include "structures.h"
#include "defines.h"
#include "functions.h"
```

Graphe des dépendances par inclusion de functions.c:

### Fonctions

- `PtrList createNode (Process p, PtrList next)`  
*Fonction créant un noeud de la file de processus.*
- `void ajout_activite (PtrList *ordonnanceurHead, PtrList *ordonnanceurTail, Process processus)`  
*Fonction ajoutant un processus à la file.*
- `void step (PtrList *ordonnanceurHead, PtrList *ordonnanceurTail)`  
*Fonction qui exécute et défile un processus.*
- `void run (PtrList ordonnanceurHead, PtrList ordonnanceurTail)`  
*Fonction exécute les processus les uns après les autres avec la fonction step.*
- `void AffListe (PtrList list)`  
*Fonction affichant la file des processus.*
- `Process registerProcess (char *name, int execution_time, int priority)`  
*Fonction créant une variable de type Process avec les données que l'on veut.*

### 4.2.1 Description détaillée

fichier des fonctions du programme

**Auteur**

Lucas Velay

**Date**

10 décembre 2022

### 4.2.2 Documentation des fonctions

#### 4.2.2.1 AffListe()

```
AffListe (
    PtrList list )
```

Fonction affichant la file des processus.

**Auteur**

Lucas Velay

**Date**

10 décembre 2022

**Paramètres**

<i>list</i>	File des processus
-------------	--------------------

#### 4.2.2.2 ajout\_activite()

```
ajout_activite (
    PtrList * ordonnanceurHead,
    PtrList * ordonnanceurTail,
    Process processus )
```

Fonction ajoutant un processus à la file.

**Auteur**

Lucas Velay

**Date**

10 décembre 2022

## Paramètres

<i>ordonnanceurHead</i>	pointeur sur la tête de la file
<i>ordonnanceurTail</i>	pointeur sur la queue de la file
<i>processus</i>	processus ajouté

## 4.2.2.3 createNode()

```
PtrList createNode (
    Process p,
    PtrList next )
```

Fonction créant un noeud de la file de processus.

## Auteur

Lucas Velay

## Date

10 décembre 2022

## Paramètres

<i>p</i>	processus ajouté à la file
<i>next</i>	valeur suivante de la liste chaîné

## Renvoie

le noeud créé

## 4.2.2.4 registerProcess()

```
registerProcess (
    char * name,
    int execution_time,
    int priority )
```

Fonction créant une variable de type `Process` avec les données que l'on veut.

## Auteur

Lucas Velay

## Date

10 décembre 2022

## Paramètres

<i>name</i>	Nom du processus
<i>execution_time</i>	Temps d'execution du processus
<i>priority</i>	Priorité du processus

## 4.2.2.5 run()

```
run (
    PtrList ordonnanceurHead,
    PtrList ordonnanceurTail )
```

Fonction execute les processus les uns après les autres avec la fonction step.

## Auteur

Lucas Velay

## Date

10 décembre 2022

## Paramètres

<i>ordonnanceurHead</i>	pointeur sur la tête de la file
<i>ordonnanceurTail</i>	pointeur sur la queue de la file

## 4.2.2.6 step()

```
step (
    PtrList * ordonnanceurHead,
    PtrList * ordonnanceurTail )
```

Fonction qui execute et defile un processus.

## Auteur

Lucas Velay

## Date

10 décembre 2022

## Paramètres

<i>ordonnanceurHead</i>	pointeur sur la tête de la file
<i>ordonnanceurTail</i>	pointeur sur la queue de la file

## 4.3 Référence du fichier src/functions.h

Fichier d'en tête des fonctions de fonctions.c.

Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

### Fonctions

- `PtrList createNode` (`Process` p, `PtrList` next)  
*Fonction créant un noeud de la file de procesus.*
- `void ajout_activite` (`PtrList` \*ordonnanceurHead, `PtrList` \*ordonnanceurTail, `Process` processus)  
*Fonction ajoutant un processus à la file.*
- `void step` (`PtrList` \*ordonnanceurHead, `PtrList` \*ordonnanceurTail)  
*Fonction qui execute et defile un processus.*
- `void run` (`PtrList` ordonnanceurHead, `PtrList` ordonnanceurTail)  
*Fonction execute les processus les uns après les autres avec la fonction step.*
- `void AffListe` (`PtrList` liste)  
*Fonction affichant la file des processus.*
- `Process registerProcess` (`char` \*name, `int` execution\_time, `int` priority)  
*Fonction créant une variable de type `Process` avec les données que l'on veut.*

### 4.3.1 Description détaillée

Fichier d'en tête des fonctions de fonctions.c.

Auteur

Lucas Velay

Date

10 décembre 2022

### 4.3.2 Documentation des fonctions

#### 4.3.2.1 AffListe()

```
void AffListe (  
    PtrList list )
```

Fonction affichant la file des processus.

Auteur

Lucas Velay

Date

10 décembre 2022

## Paramètres

<i>list</i>	File des processus
-------------	--------------------

## 4.3.2.2 ajout\_activite()

```
void ajout_activite (
    PtrList * ordonnanceurHead,
    PtrList * ordonnanceurTail,
    Process processus )
```

Fonction ajoutant un processus à la file.

## Auteur

Lucas Velay

## Date

10 décembre 2022

## Paramètres

<i>ordonnanceurHead</i>	pointeur sur la tête de la file
<i>ordonnanceurTail</i>	pointeur sur la queue de la file
<i>processus</i>	processus ajouté

## 4.3.2.3 createNode()

```
PtrList createNode (
    Process p,
    PtrList next )
```

Fonction créant un noeud de la file de processus.

## Auteur

Lucas Velay

## Date

10 décembre 2022

## Paramètres

<i>p</i>	processus ajouté à la file
<i>next</i>	valeur suivante de la liste chaîné



**Renvoie**

le noeud créé

**4.3.2.4 registerProcess()**

```
Process registerProcess (
    char * name,
    int execution_time,
    int priority )
```

Fonction créant une variable de type `Process` avec les données que l'on veut.

**Auteur**

Lucas Velay

**Date**

10 décembre 2022

**Paramètres**

<i>name</i>	Nom du processus
<i>execution_time</i>	Temps d'exécution du processus
<i>priority</i>	Priorité du processus

**4.3.2.5 run()**

```
void run (
    PtrList ordonnanceurHead,
    PtrList ordonnanceurTail )
```

Fonction exécute les processus les uns après les autres avec la fonction step.

**Auteur**

Lucas Velay

**Date**

10 décembre 2022

**Paramètres**

<i>ordonnanceurHead</i>	pointeur sur la tête de la file
<i>ordonnanceurTail</i>	pointeur sur la queue de la file

#### 4.3.2.6 step()

```
void step (
    PtrList * ordonnanceurHead,
    PtrList * ordonnanceurTail )
```

Fonction qui exécute et défiler un processus.

##### Auteur

Lucas Velay

##### Date

10 décembre 2022

##### Paramètres

<i>ordonnanceurHead</i>	pointeur sur la tête de la file
<i>ordonnanceurTail</i>	pointeur sur la queue de la file

## 4.4 Référence du fichier src/main.c

Main program.

```
#include <stdio.h>
#include <stdlib.h>
#include "structures.h"
#include "defines.h"
#include "functions.h"
```

Graphe des dépendances par inclusion de main.c:

### Fonctions

— int [main](#) (int argc, char const \*argv[])  
*Main function.*

#### 4.4.1 Description détaillée

Main program.

##### Auteur

Lucas Velay

##### Date

10 décembre 2022

## 4.4.2 Documentation des fonctions

### 4.4.2.1 main()

```
int main (
    int argc,
    char const * argv[] )
```

Main function.

#### Auteur

Lucas Velay

#### Date

10 décembre 2022

#### Paramètres

<i>argc</i>	nombre d'arguments en entrée
<i>argv</i>	valeurs des arguments en entrée

#### Renvoie

0 si tout s'est bien passé

## 4.5 Référence du fichier src/structures.h

Fichier de structures.

Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

### Classes

- struct [defProcess](#)
- struct [defList](#)

### Définitions de type

- typedef struct [defProcess](#) **Process**
- typedef struct [defList](#) **List**
- typedef [List](#) \* **PtrList**

### 4.5.1 Description détaillée

Fichier de structures.

Auteur

Lucas Velay

Date

10 décembre 2022

# Index

AffListe  
    functions.c, [8](#)  
    functions.h, [11](#)  
ajout\_activite  
    functions.c, [8](#)  
    functions.h, [12](#)  
  
createNode  
    functions.c, [9](#)  
    functions.h, [12](#)  
  
defList, [5](#)  
    next, [5](#)  
    processus, [5](#)  
defProcess, [6](#)  
    execution\_time, [6](#)  
    name, [6](#)  
    priority, [6](#)  
  
execution\_time  
    defProcess, [6](#)  
  
functions.c  
    AffListe, [8](#)  
    ajout\_activite, [8](#)  
    createNode, [9](#)  
    registerProcess, [9](#)  
    run, [10](#)  
    step, [10](#)  
functions.h  
    AffListe, [11](#)  
    ajout\_activite, [12](#)  
    createNode, [12](#)  
    registerProcess, [13](#)  
    run, [13](#)  
    step, [14](#)  
  
List, [6](#)  
  
main  
    main.c, [15](#)  
main.c  
    main, [15](#)  
  
name  
    defProcess, [6](#)  
next  
    defList, [5](#)  
  
priority  
    defProcess, [6](#)  
  
Process, [6](#)  
processus  
    defList, [5](#)  
  
registerProcess  
    functions.c, [9](#)  
    functions.h, [13](#)  
  
run  
    functions.c, [10](#)  
    functions.h, [13](#)  
  
src/defines.h, [7](#)  
src/functions.c, [7](#)  
src/functions.h, [11](#)  
src/main.c, [14](#)  
src/structures.h, [15](#)  
step  
    functions.c, [10](#)  
    functions.h, [14](#)