

# Introduction à la programmation concurrente

Changement de contexte

---

Yann Thoma, Fiorenzo Gamba

Février 2012

Reconfigurable and Embedded Digital Systems Institute  
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0; ←
10    while(i<100) {
11        i=i+1;
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16     int i=0; ←
17     while(i<100) {
18         i=i+1;
19         sleep(1);
20     }
21     return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1		X	
T2	X		
T3	X		

Tâche 1

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) {
4         i=i+1; ←
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0; ←
10    while(i<100) {
11        i=i+1;
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16    int i=0; ←
17    while(i<100) {
18        i=i+1;
19        sleep(1);
20    }
21    return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1		X	
T2	X		
T3	X		

Tâche 1

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0; ←
10    while(i<100) {
11        i=i+1;
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16     int i=0; ←
17     while(i<100) {
18         i=i+1;
19         sleep(1);
20     }
21     return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1		X	
T2	X		
T3	X		

Tâche 1

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) {
4         i=i+1; ←
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0; ←
10    while(i<100) {
11        i=i+1;
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16    int i=0; ←
17    while(i<100) {
18        i=i+1;
19        sleep(1);
20    }
21    return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1		X	
T2	X		
T3	X		

Tâche 1

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0; ←
10    while(i<100) {
11        i=i+1;
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16     int i=0; ←
17     while(i<100) {
18         i=i+1;
19         sleep(1);
20     }
21     return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1		X	
T2	X		
T3	X		

Préemption de T1 par T2

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0; ←
10    while(i<100) {
11        i=i+1;
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16    int i=0; ←
17    while(i<100) {
18        i=i+1;
19        sleep(1);
20    }
21    return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1	X		
T2		X	
T3	X		

Tâche 2

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) { ←
11        i=i+1;
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16    int i=0; ←
17    while(i<100) {
18        i=i+1;
19        sleep(1);
20    }
21    return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1	X		
T2		X	
T3	X		

Tâche 2



## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) {
11        i=i+1; ←
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16     int i=0; ←
17     while(i<100) {
18         i=i+1;
19         sleep(1);
20     }
21     return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1	X		
T2		X	
T3	X		

Tâche 2

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) { ←
11        i=i+1;
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16    int i=0; ←
17    while(i<100) {
18        i=i+1;
19        sleep(1);
20    }
21    return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1	X		
T2		X	
T3	X		

Tâche 2

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) {
11        i=i+1; ←
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16     int i=0; ←
17     while(i<100) {
18         i=i+1;
19         sleep(1);
20     }
21     return NULL;
22 }
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1	X		
T2		X	
T3	X		

Préemption de T2 par T3

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) {
11        i=i+1; ←
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16     int i=0; ←
17     while(i<100) {
18         i=i+1;
19         sleep(1);
20     }
21     return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1	X		
T2	X		
T3		X	

Tâche 3

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) {
11        i=i+1; ←
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16    int i=0;
17    while(i<100) { ←
18        i=i+1;
19        sleep(1);
20    }
21    return NULL;
22 }
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1	X		
T2	X		
T3		X	

Tâche 3

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) {
11        i=i+1; ←
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16     int i=0;
17     while(i<100) {
18         i=i+1; ←
19         sleep(1);
20     }
21     return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1	X		
T2	X		
T3		X	

Tâche 3

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) {
11        i=i+1; ←
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16     int i=0;
17     while(i<100) {
18         i=i+1;
19         sleep(1); ←
20     }
21     return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1	X		
T2	X		
T3		X	

T3 passe en attente

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) {
11        i=i+1; ←
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16    int i=0;
17    while(i<100) {
18        i=i+1;
19        sleep(1); ←
20    }
21    return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1		X	
T2	X		
T3			X

Tâche 1



## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) {
4         i=i+1; ←
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) {
11        i=i+1; ←
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16    int i=0;
17    while(i<100) {
18        i=i+1;
19        sleep(1); ←
20    }
21    return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1		X	
T2	X		
T3			X

Tâche 1

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) {
11        i=i+1; ←
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16     int i=0;
17     while(i<100) {
18         i=i+1;
19         sleep(1); ←
20     }
21     return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1		X	
T2	X		
T3			X

Tâche 1

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) {
4         i=i+1; ←
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) {
11        i=i+1; ←
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16    int i=0;
17    while(i<100) {
18        i=i+1;
19        sleep(1); ←
20    }
21    return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1		X	
T2	X		
T3			X

Tâche 1

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) {
11        i=i+1; ←
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16     int i=0;
17     while(i<100) {
18         i=i+1;
19         sleep(1); ←
20     }
21     return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1		X	
T2	X		
T3			X

Préemption de T1 par T2

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) {
11        i=i+1; ←
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16    int i=0;
17    while(i<100) {
18        i=i+1;
19        sleep(1); ←
20    }
21    return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1	X		
T2		X	
T3			X

Tâche 2

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) { ←
11        i=i+1;
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16    int i=0;
17    while(i<100) {
18        i=i+1;
19        sleep(1); ←
20    }
21    return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1	X		
T2		X	
T3			X

Tâche 2

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) {
11        i=i+1; ←
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16    int i=0;
17    while(i<100) {
18        i=i+1;
19        sleep(1); ←
20    }
21    return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1	X		
T2		X	
T3			X

Tâche 2

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) { ←
11        i=i+1;
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16    int i=0;
17    while(i<100) {
18        i=i+1;
19        sleep(1); ←
20    }
21    return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1	X		
T2		X	
T3			X

Préemption de T2 par T1



## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) { ←
11        i=i+1;
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16     int i=0;
17     while(i<100) {
18         i=i+1;
19         sleep(1); ←
20     }
21     return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1		X	
T2	X		
T3			X

Tâche 1

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) {
4         i=i+1; ←
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) { ←
11        i=i+1;
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16    int i=0;
17    while(i<100) {
18        i=i+1;
19        sleep(1); ←
20    }
21    return NULL;
22 }
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1		X	
T2	X		
T3			X

Tâche 1

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) { ←
11        i=i+1;
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16     int i=0;
17     while(i<100) {
18         i=i+1;
19         sleep(1); ←
20     }
21     return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1		X	
T2	X		
T3			X

1 seconde s'est écoulée

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) { ←
11        i=i+1;
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16     int i=0;
17     while(i<100) { ←
18         i=i+1;
19         sleep(1);
20     }
21     return NULL;
22 }
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1	X		
T2	X		
T3		X	

Tâche 3

## Exemple

```
1 void *tache1(void *args) {
2     int i=0;
3     while(i<100) { ←
4         i=i+1;
5     }
6     return NULL;
7 }
8 void *tache2(void *args) {
9     int i=0;
10    while(i<100) { ←
11        i=i+1;
12    }
13    return NULL;
14 }
15 void *tache3(void *args) {
16    int i=0;
17    while(i<100) {
18        i=i+1; ←
19        sleep(1);
20    }
21    return NULL;
```

← PC courant

← PC sauvegardé

	prête	élue	bloquée
T1	X		
T2	X		
T3		X	

Tâche 3