



المعهد الوطني للبريد والهواصالات
መግቢያ ስልክ ለግብይት ለግብይት ለግብይት
Institut National des Postes et Télécommunications

INSITUT NATIONAL DES POSTES ET TELECOMMUNICATIONS

ATELIERS D'INGÉNIERIE PROJET
RAPPORT

Simulateur TokenRing

Realisé par :

Ibrahim AJAOUN

Badr LACHHEB

Imad Eddine BEL-HOUIDEG

20 janvier 2025

Table des matières

1	Introduction	2
2	Presentation du Token Ring	2
3	Description de l'implémentation	2
3.1	Gestion du Jeton avec un Semaphore	2
3.2	Simulation du Reseau	2
4	Code Source	2
5	Tests et Resultats	4
6	Conclusion	4

1 Introduction

Le but de ce projet est de développer un simulateur de réseau **Token Ring**, une topologie de communication utilisée dans les réseaux locaux (LAN). Cette simulation reproduit le fonctionnement du protocole où un jeton circule entre les noeuds pour gérer l'accès au réseau.

2 Presentation du Token Ring

Le protocole **Token Ring** permet une communication structurée entre plusieurs machines en utilisant un jeton qui circule de manière unidirectionnelle. Un noeud ne peut émettre des données que s'il possède ce jeton. Si un noeud est en panne, le système doit gérer cette situation pour assurer la continuité du réseau.

3 Description de l'implémentation

3.1 Gestion du Jeton avec un Semaphore

Un **semaphore** est utilisé afin de représenter le jeton. Son rôle est de garantir qu'un seul noeud à la fois peut émettre des données.

- `down(struct Semaphore* s)` : Vérifie si le jeton est disponible et le capture.
- `up(struct Semaphore* s)` : Libère le jeton pour le prochain noeud.

3.2 Simulation du Réseau

- Le programme demande à l'utilisateur de saisir le **nombre de noeuds** (limité à 10).
- Chaque noeud peut être **actif** ou **en panne**, un état déterminé aléatoirement.
- La simulation fonctionne en boucle infinie :
 - Un noeud actif acquiert le jeton et effectue un **traitement**.
 - Un noeud peut **tomber en panne** avec une probabilité de 20%.
 - Le jeton est ensuite transmis au **noeud suivant**.

4 Code Source

Listing 1 – Implementation du Token Ring

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <unistd.h>

struct Semaphore {
    int value;
};
```

```
void init_semaphore(struct Semaphore* s, int value) {
    s->value = value;
}

void down(struct Semaphore* s) {
    while (s->value <= 0) {
    }
    s->value--;
}

void up(struct Semaphore* s) {
    s->value++;
}

void simulate_token_ring(int num_nodes) {
    struct Semaphore token;
    init_semaphore(&token, 1);
    int current_node = 0;
    int node_status[num_nodes]; //   tat   des n uds (1 = actif,
                                //   0 = en panne)

    printf("Initialisation_des_etats_des_noeuds_\n");
    for (int i = 0; i < num_nodes; i++) {
        node_status[i] = rand() % 2;
        printf("Noeud_%d_:_%s\n", i, node_status[i] ? "Actif" :
            "En_panne");
    }

    while (1) {
        while (node_status[current_node] == 0) {
            printf("Le_noeud_%d_est_en_panne._On_passe_au_
                suivant...\n", current_node);
            current_node = (current_node + 1) % num_nodes;
        }
        down(&token);
        printf("Le_noeud_%d_a_recu_le_jeton\n", current_node);

        printf("Le_noeud_%d_traite_ses_donnees\n", current_node)
            ;

        // Decider aleatoirement si le noeud tombe en panne
        if (rand() % 10 < 2) {
            // 20% de chance de panne
            printf("Le_noeud_%d_tombe_en_panne\n", current_node)
                ;
            node_status[current_node] = 0;
        }
    }
}
```

```
    }

    printf("Le_noeud_%d_passe_le_jeton\n", current_node);
    current_node = (current_node + 1) % num_nodes;
    up(&token);
}

int main() {
    int num_nodes;
    srand(time(NULL));

    printf("Entrez le_nombre_de_noeuds_dans_le_reseau_(max_10):
    _");
    scanf("%d", &num_nodes);

    if (num_nodes <= 0 || num_nodes > 10) {
        printf("Nombre_de_noeuds_invalide\n");
        return 1;
    }

    simulate_token_ring(num_nodes);
    return 0;
}
```

5 Tests et Resultats

Lors des tests, on observe :

- Une bonne circulation du jeton entre les noeuds actifs.
- Une gestion correcte des **pannes** en sautant les noeuds defectueux.
- Une simulation realiste du comportement d'un reseau **Token Ring**.

6 Conclusion

Ce projet nous fournit une implémentation fonctionnelle du **Token Ring**, respectant l'algorithme donne. Il met en evidence les avantages et les limites de cette topologie, en particulier la gestion des pannes et la transmission controlée du jeton.