

PTS FINAL

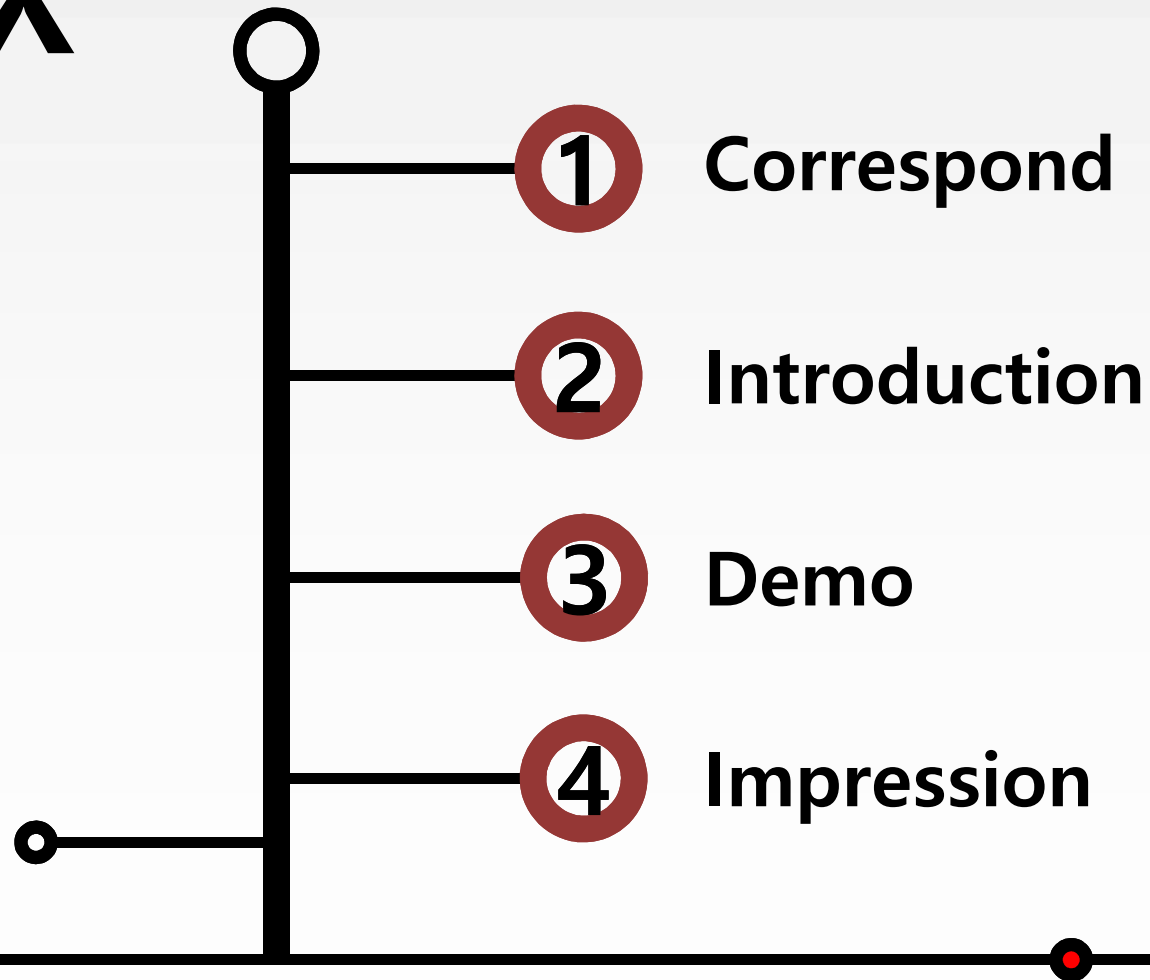
T1 Team

김용현

김준호

박정민

INDEX



CORRESPOND



CORRESPOND

Identifier	Feature	Valid / Invalid value	Check
1	(승차) 지하철->버스 환승	잔액이 500원 이상일 경우에 승차 여부 & 결제 금액, 잔액, 시간 출력 및 저장 확인	잔액이 마이너스가 여도 승차 가능 ->PASS

Identifier	Feature	Valid / Invalid value	Check
11	3분 정산	단말기의 정보 초기화 확인	정산 시 버스파일 초기화 불가 -> PASS

CORRESPOND

Identifier	Feature	Valid / Invalid value	Check
1	(승차) 버스->지하철 환승	잔액이 600원 이상일 경우에 승차 여부 & 결제 금액, 잔액, 시간 출력 및 저장 확인	잔액이 1050원 미만에 환승 불가 ->PASS

Identifier	Feature	Valid / Invalid value	Check
6	(하차) 버스->지하철 환승	정거장당 300원 요금 계산 여부	1정거장 시 0원 2정거장 시 300원 ->PASS

CORRESPOND

Identifier	Feature	Valid / Invalid value	Check
1	최초 탑승	하차 태그가 불가능한지 확인	하차 가능
2	버스 or 지하철 태그 하차 후	하차 태그가 불가능한지 확인	하차 가능
3	버스 승차 후	하차 태그가 불가능한지 확인	하차 가능
4	미정산 초기화 후	하차 태그가 불가능한지 확인	하차 가능

Identifier	Feature	Valid / Invalid value	Check
1	최초 탑승	하차 태그가 불가능한지 확인	하차 가능
2	버스 or 지하철 태그 하차 후	하차 태그가 불가능한지 확인	하차 가능
3	버스 승차 후	하차 태그가 불가능한지 확인	하차 가능
4	미정산 초기화 후	하차 태그가 불가능한지 확인	하차 가능

CORRESPOND

Identifier	Feature	Valid / Invalid value	Check
1	3분 정산	버스단말기 정보와 지하철 단말기들의 정보를 읽은 후 정해진 계산식에 따라 계산 후 버스 및 지하철 회사로 출력 및 전송 확인	정산 시 버스파일 초기화 안 된다. 정산은 제대로 되지 않는다.

CORRESPOND

BUS

Identifier 3,4

METRO

Identifier 2,3,4,5

이 부분에 대해 버스 요금규정과 지하철 요금 규정을 각 단말기 별로 처리해야 하는 부분이어서 1850, 1950원과 같은 금액은 나올 수 없다고 본다.

미 정산 후 추가요금을 다른 회사 단말기에 추가하는 것이 이상하다고 생각한다.

Ex> 지하철 미 정산 후 버스를 환승 없이 탄다

T4의 의견 : 1250원이 버스에 부과된다.

T1의 반론 : 1050원이 버스에 부과된다. 지하철 미 정산 요금 200원은 정산에서 지하철에 전달 될 수 없다.

CORRESPOND

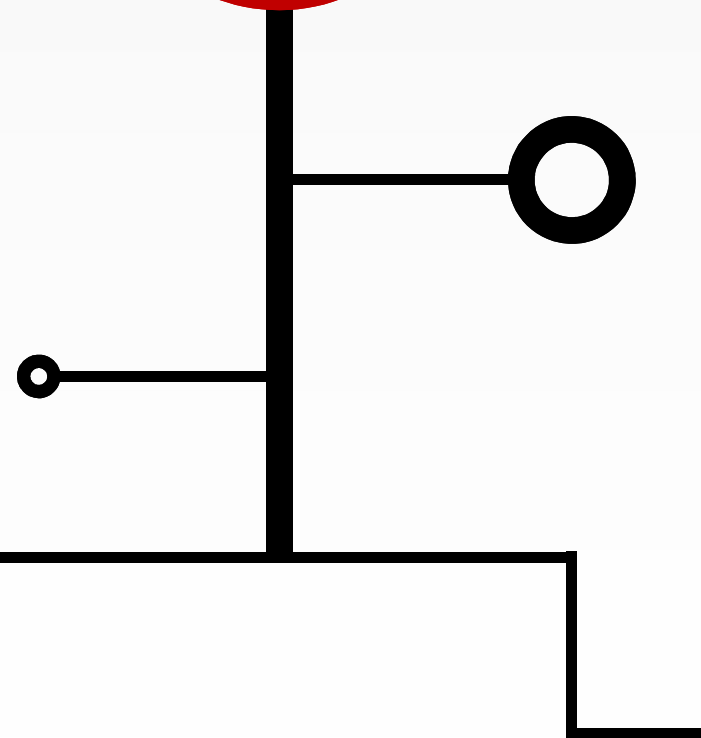
EXCEPTION

예외와 관련된 부분 중 하차 후 하차와 관련되어 있는 경우는 시뮬레이션으로 만든 것이므로 연속된 OUT과 관련한 사항을 고려하지 않는다. 또한 다른 교통수단에서 OUT 역시 고려하지 않는다.

실생활에서 보면 카드를 태그 하는 것만 있을 뿐 IN OUT을 설정하지 않는다.

이는 단순히 시뮬레이션을 위해 만들어진 것이며, SRS에서 고려하는 미 정산을 제외하고는 IN OUT의 순서로의 입력만을 본다.

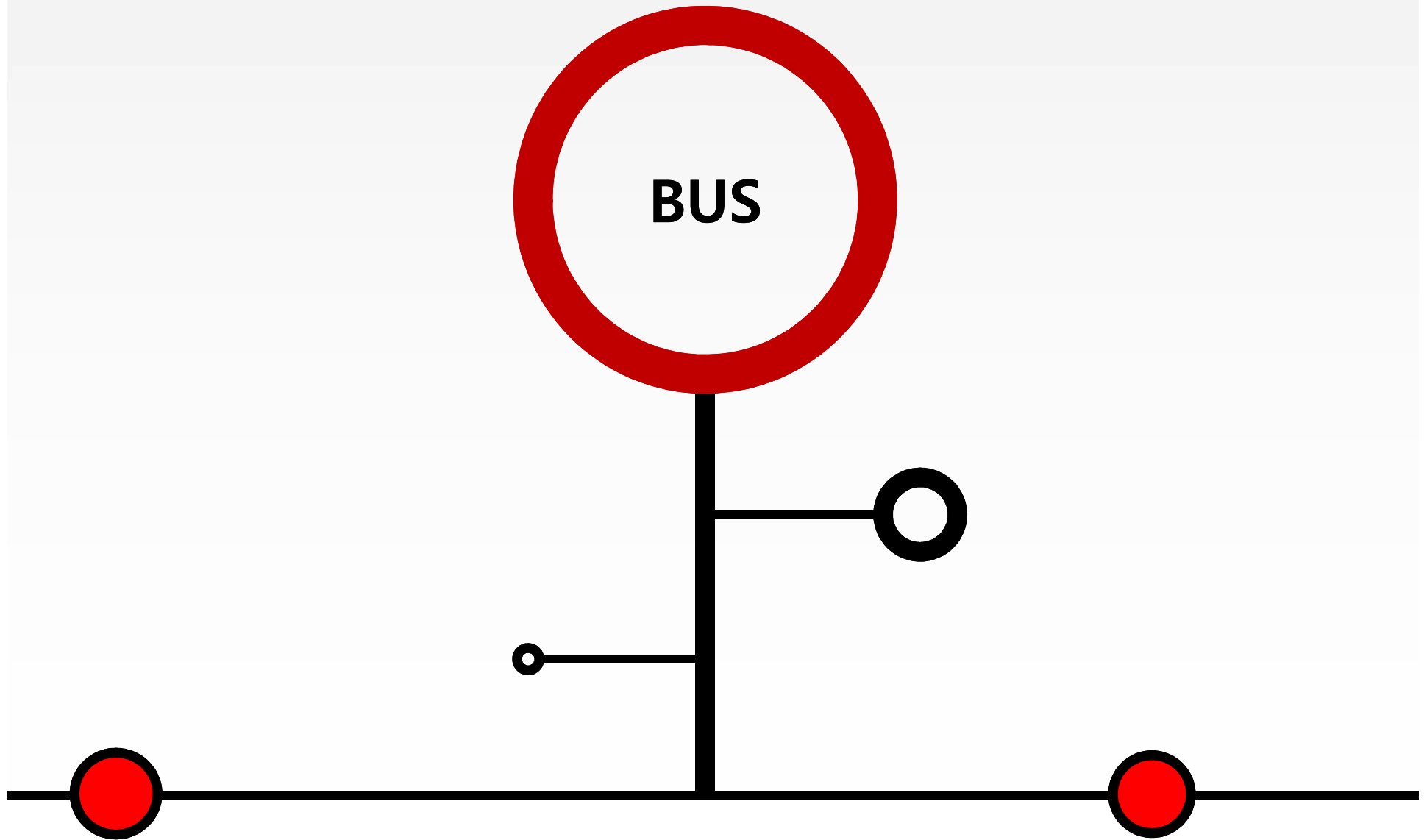
INTRODUCTION



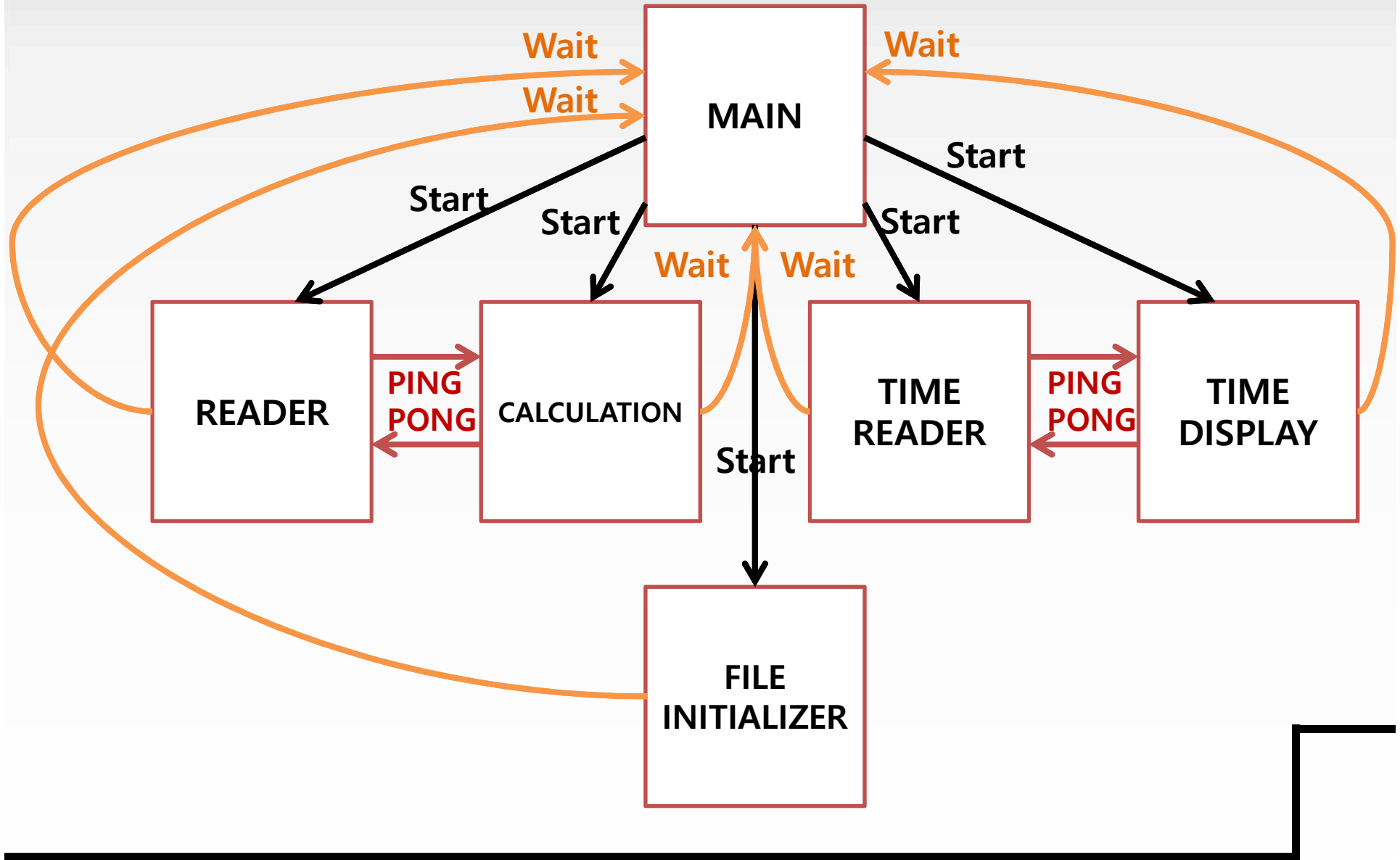
INTRODUCTION



INTRODUCTION



INTRODUCTION



INTRODUCTION

```
8 #define IDENTIFIER 6
9 typedef struct bustime{
10     int min;
11     int hour;
12     int day;
13     int month;
14     int year;
15 }BUSTIME;
16
17 typedef struct card{
18     BUSTIME time;
19     char* traffic;
20     int oldInout;
21     int newInout;
22     int balance;
23     char* IDcount;
24     int fee;
25     char* filename;
26 }CARDINFO;
```

버스 식별자

버스 시간 관리
구조체

고객정보 관리
구조체

INTRODUCTION

```
pthread_mutex_t sync_mutex;  
pthread_mutex_t gmutex;  
pthread_cond_t sync_cond;  
pthread_cond_t gcond;  
pthread_mutex_t disp_mutex;  
pthread_mutex_t read_mutex;  
pthread_cond_t disp_cond;  
pthread_cond_t read_cond;
```

**PING-PONG을 위한
Mutex & Cond
관계변수들 선언**

Sync_cond : reader의 권한

G_cond : calculation의 권한

Disp_cond : time display의 권한

Read_cond : time reader의 권한

INTRODUCTION

```
void parser();  
void* reader(void* arg);  
void* timereader(void* arg);  
void display(int display_Command);  
void readerWriteController(int readerWriteCommand);  
void cardWriteController(int cardwritecommand);  
void* time_Display(void* arg);
```

Void* reader(void* arg) : Command를 읽어온다.

Void parser() : 카드 데이터를 파싱한다.

Void* timereader(void* arg) : Time.txt의 시간을 읽어온다.

Void display(int display_Command) : Display에 대한 함수호출

Void readerWriteController(int readerWriteCommand) : 리더기 파일 관련 함수호출

Void cardWriteController(int cardwritecommand) : 카드 파일 관련 함수호출

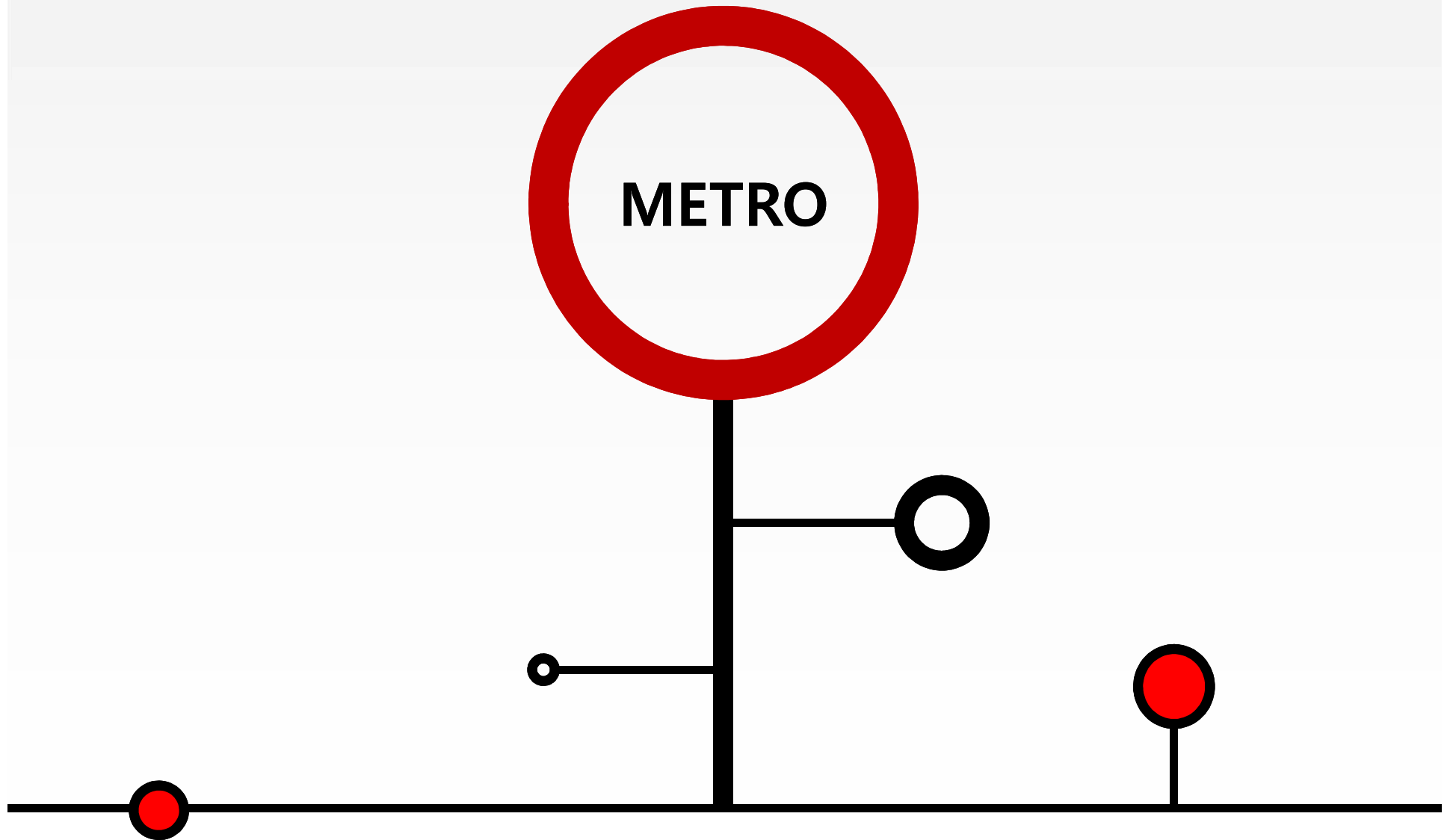
Void* time_display(void* arg) : 시간을 초마다 보여주기위한 함수

INTRODUCTION

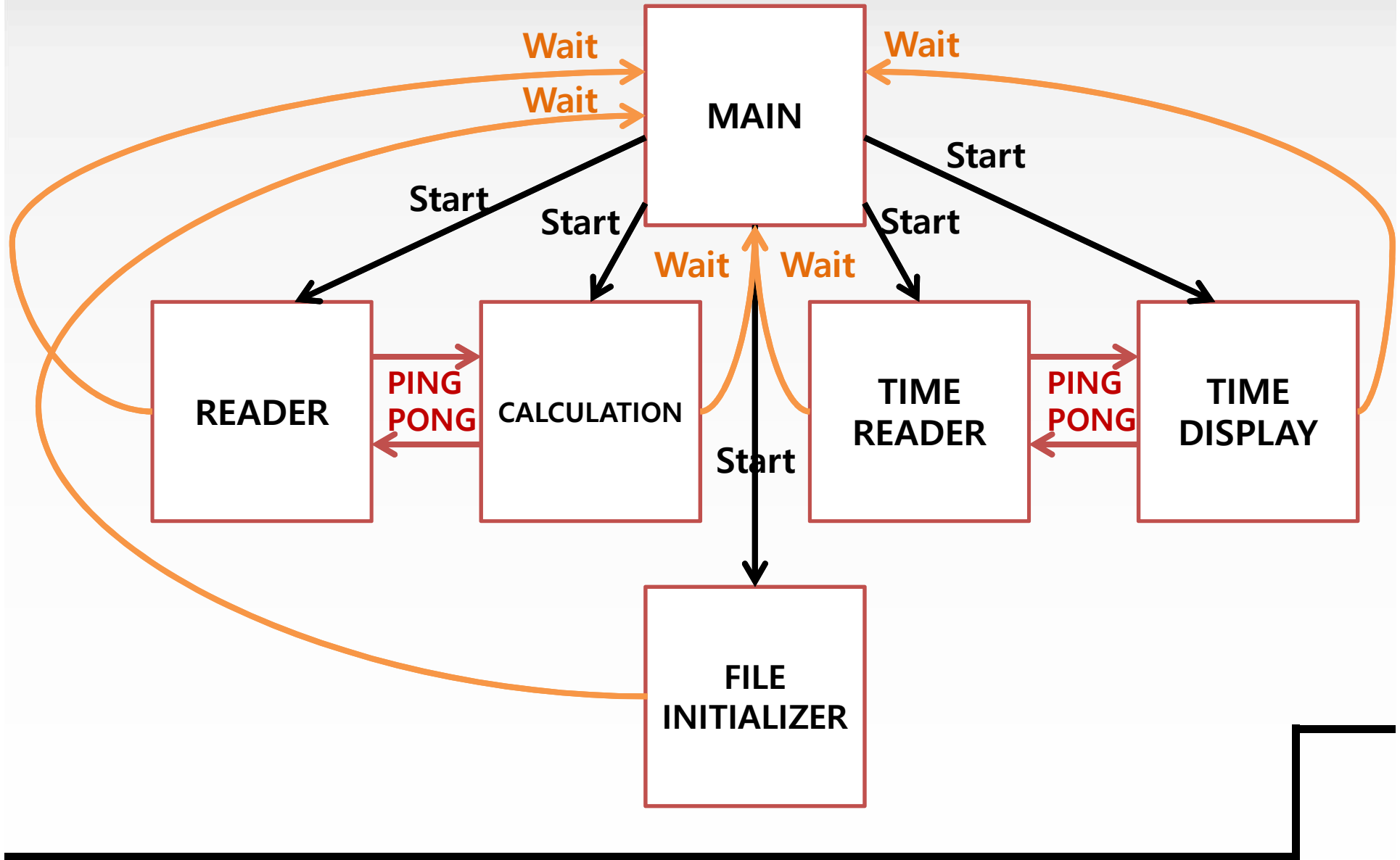
```
void* Calculate_Controller(void* arg);  
void Standard_Fee_Controller();  
void Standard_Fee_in();  
void Standard_Fee_out();  
void Transfer_Fee_Controller();  
void Transfer_Fee_in();  
void Transfer_Fee_out();  
void Transfer_NonCalculate_Fee_in();  
void Lack_Money();  
int timeLag(BUSTIME time);  
int timeChange(BUSTIME time);  
void* fileInitializer(void* arg);
```

Void* Calculate_Controller(void * arg) : 계산관련 함수 호출
Void Standard_Fee_Controller() : 기본요금 계산관련 함수 호출
Void Standard_Fee_in() : 기본요금 IN
Void Standard_Fee_out() : 기본요금 OUT
Void Transfer_Fee_Controller() : 환승요금
Void Transfer_Fee_in() : 환승 IN 요금
Void Transfer_Fee_out() : 환승 OUT 요금
Void Lack_Money() : 요금부족 화면출력
int timeLag(BUSTIME time) : 시간차 계산 함수
int timeChange(BUSTIME time) : 시간 변환 함수
Void* fileInitializer(void* arg) : 파일 초기화 함수

INTRODUCTION



INTRODUCTION



INTRODUCTION

```
typedef struct metrotime{  
    int min;  
    int hour;  
    int day;  
    int month;  
    int year;  
}METROTIME;
```

시간 관리 구조체

```
typedef struct card{  
    METROTIME time;  
    char* traffic;  
    int oldInout;  
    int newInout;  
    int balance;  
    char* IDcount;  
    int fee;  
    char* filename;  
}CARDINFO;
```

카드 파일 관리 구조체

INTRODUCTION

```
pthread_mutex_t sync_mutex;  
pthread_mutex_t gmutex;  
pthread_cond_t sync_cond;  
pthread_cond_t gcond;
```

```
pthread_mutex_t disp_mutex;  
pthread_mutex_t read_mutex;  
pthread_cond_t disp_cond;  
pthread_cond_t read_cond;
```

**PING-PONG을 위한
Mutex & Cond
관계변수들 선언**

Sync_cond : reader의 권한

G_cond : calculation의 권한

Disp_cond : time display의 권한

Read_cond : time reader의 권한

INTRODUCTION

```
void parser();  
void* reader(void* arg);  
void* timereader(void* arg);  
void display(int display_Command);  
void readerWriteController(int readerWriteCommand);  
void cardWriteController(int cardWriteCommand);  
void* fileInitializer(void* arg);  
void* time_Display(void* arg);
```

Void parser() : 카드 파일 데이터를 나누어 구조체에 담는다.

Void* reader(void* arg) : Command를 읽어온다.

Void* timereader(void* arg) : Time.txt 파일의 시간을 읽어온다.

Void display(int display_Command) : Display관련 함수호출

Void readerWriteController(int readerWriteCommand) : 리더기 파일
관련 함수 호출

Void cardWriteController(int cardWriteCommand) : 카드파일 관련
함수호출

Void* fileInitializer(void* arg) : 파일 초기화

Void* time_Display(void* arg) : 시간을 보여주는 함수

INTRODUCTION

```
void* Calculate_Controller(void* arg);  
void Standard_Fee_Controller();  
void Standard_Fee_in();  
void Standard_Fee_out();  
void Additional_Fee_out();  
void NonCalculate_Fee_in();  
void Transfer_Fee_Controller();  
void Transfer_Fee_in();  
void Transfer_Fee_out();  
void Transfer_NonCalculate_Fee_in();  
void Lack_Money();  
int timeLag(METROTIME time);  
int timeChange(METROTIME time);  
int distance();
```

계산관련 함수 호출
기본요금 관련 함수 호출
기본요금 IN
기본요금 OUT
추가요금 OUT
미정산요금 IN
환승요금 관련 함수 호출
환승 IN
환승 OUT
환승 미정산 IN
요금부족
시간차 계산
시간 변환
거리 계산

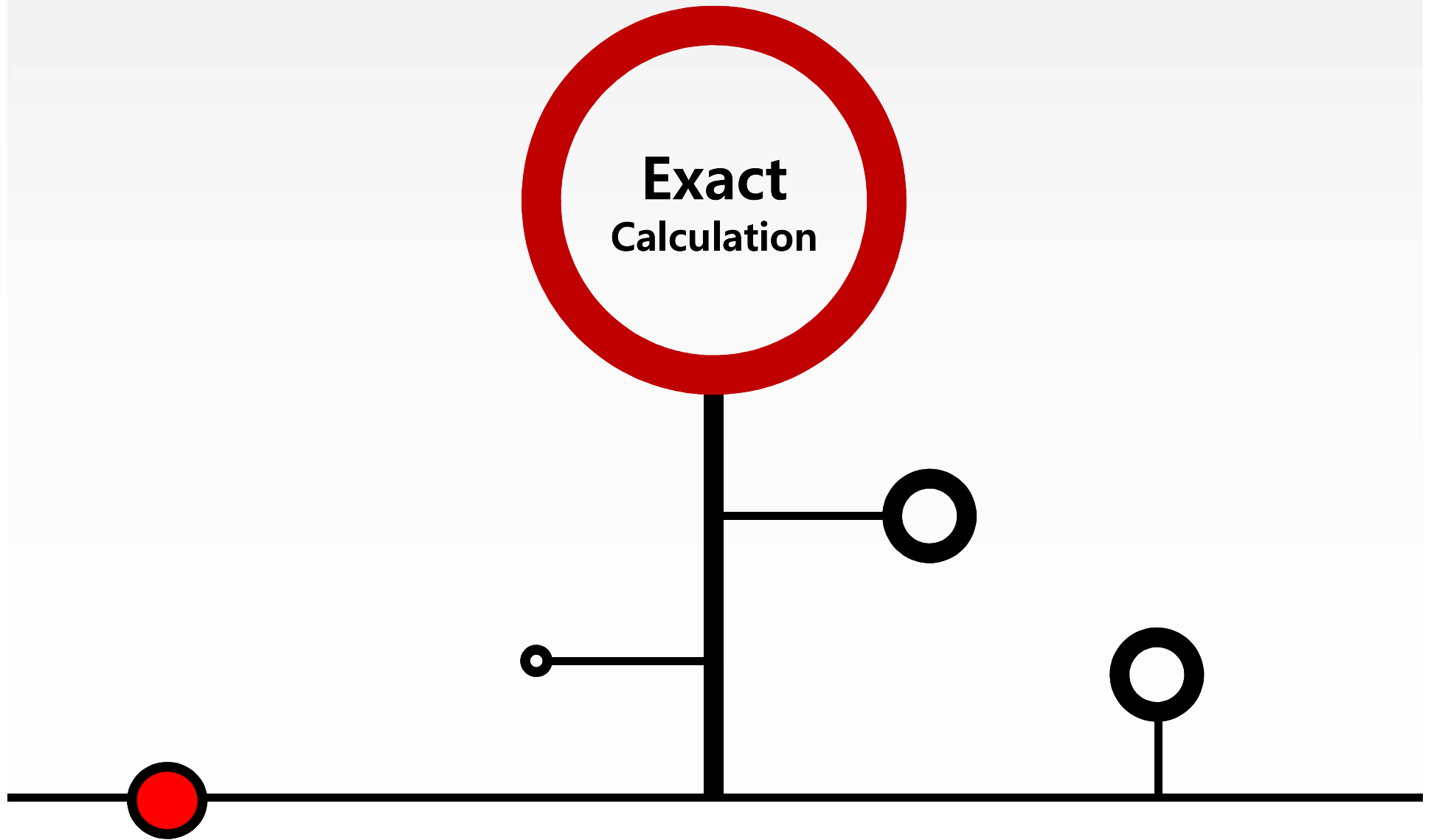
INTRODUCTION

```
int locationLoad();  
void locationWrite();
```

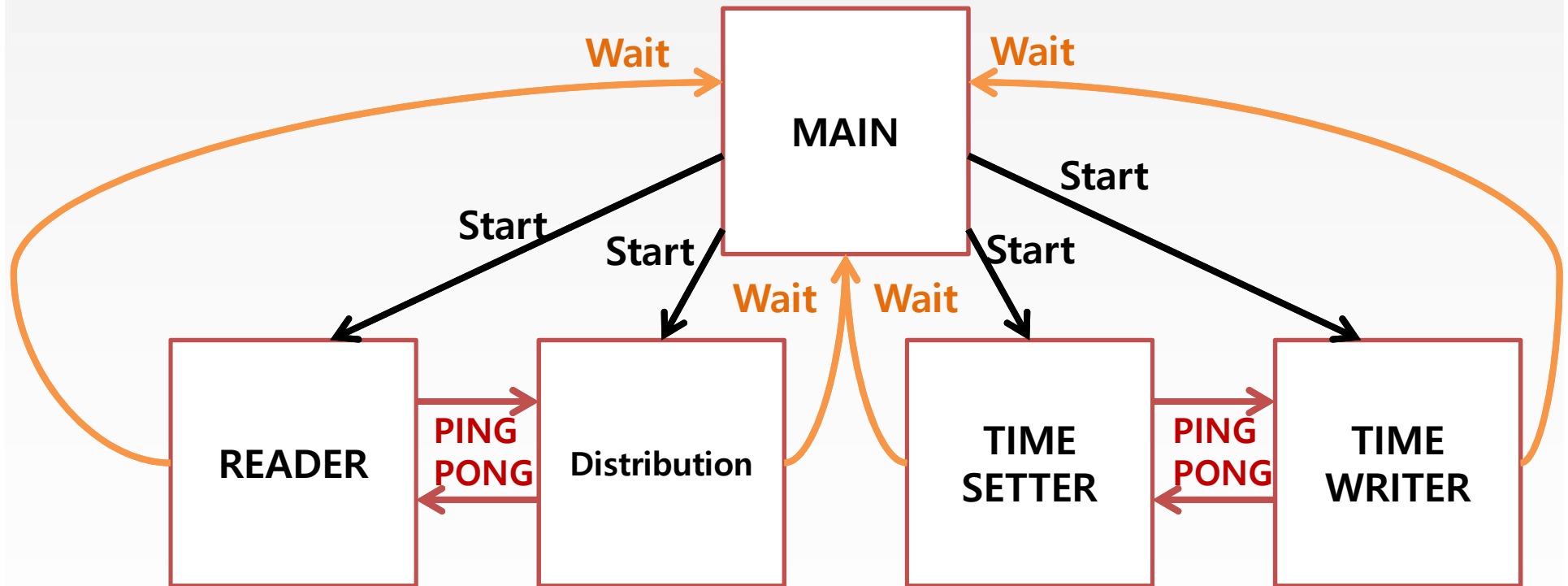
환승 후 거리당 추가요금 부과 계산을 위해 만든 함수

Void locationWrite() : location.txt파일에 ID를 쓴다.
Int locationLoad() : location.txt 파일에 ID를 읽어와
Integer형으로 변환하여 전달한다.

INTRODUCTION



INTRODUCTION



INTRODUCTION

```
typedef struct exactcalculated{  
    int busresult;  
    int metroresult;  
}EXACTCALCULATED;
```

정산 계산 저장

```
typedef struct ptstime{  
    int year;  
    int month;  
    int day;  
    int hour;  
    int min;  
}PTSTIME;
```

전체 시간을 관리
하는 구조체

```
typedef struct node NODE;  
struct node{  
    int time;  
    char* traffic;  
    int inout;  
    int fee;  
    char* IDcount;  
    NODE* next;  
    NODE* prev;  
};
```

파일에 있는 자료
를 담은 NODE

INTRODUCTION

```
pthread_mutex_t sync_mutex;  
pthread_mutex_t gmutex;  
pthread_mutex_t time_mutex;  
pthread_mutex_t write_mutex;  
pthread_cond_t sync_cond;  
pthread_cond_t gcond;  
pthread_cond_t time_cond;  
pthread_cond_t write_cond;
```

**PING-PONG을 위한
Mutex & Cond
관계변수들 선언**

Sync_cond : reader의 권한

G_cond : distribution의 권한

time_cond : timer 의 권한

write_cond : time writer의 권한

INTRODUCTION

```
void display_Controller(int displayCommand);  
void exact_CalculationDisplay();  
NODE* addNode(char* data);  
NODE* deleteNode(NODE* temp);  
NODE** find(char* idcount);  
void deleteAll();  
void print();
```

Void display_Controller(int displayCommad) : 정산결과 출력
함수 호출

Void exact_CalculationDisplay() : 정산결과 출력

NODE* addNode(char* data) : 노드를 추가하면서 시간순으로
정렬

NODE* deleteNode(NODE* temp) : 노드 삭제

NODE** find(char* temp) : data를 통해 노드 검색

Void deleteAll() : 전체 노드 삭제

Void print() : 전체 노드의 data중 시간 출력

INTRODUCTION

```
void* distribute_Controller(void* arg);  
void transferfilter();  
void busfilter();  
void metrofilter();  
void* reader(void* arg);  
void* time_setter(void* arg);  
void* time_writer(void* arg);
```

Void* distribute_Controller(void* arg) : 정산 관련 함수 호출
Void transferfilter() : 환승관련 노드만 계산후 노드삭제
Void busfilter() : 버스관련 노드만 계산
Void metrofilter() : 지하철관련 노드만 계산후 노드 전체 삭제, 화면 출력 함수 호출
Void* reader(void* arg) : 정산 관련 파일읽어오는 함수
Void* time_setter(void* arg) : 시간 설정함수 1초당 8분
Void* time_writer(void* arg) : time.txt파일에 시간을 쓰는 함수

DEMO



DEMO

```
COMMAND LINE_:2014/ 12/ 11/ 2/ 24
2014/ 12/ 11/ 2/ 24
abc I2014/ 12/ 11/ 2/ 32
N
Cardfile Write !
Readefile Write!
FEE : 1050
UALANCE : 500
COMMAND LINE_:2014/ 12/ 11/ 2/ 40
ab2014/ 12/ 11/ 2/ 48
c OUT2014/ 12/ 11/ 2/ 56

Cardfile Write !
Readefile Write!
FEE : 0
UALANCE : 500
COMMAND LINE_:2014/ 12/ 11/ 3/ 4
2014/ 12/ 11/ 3/ 12
```

```
2014/ 12/ 11/ 2/ 32
2014/ 12/ 11/ 2/ 40
2014/ 12/ 11/ 2/ 48
2014/ 12/ 11/ 2/ 56
2014/ 12/ 11/ 3/ 4
2014/ 12/ 11/ 3/ 12
2014/ 12/ 11/ 3/ 20
abc2014/ 12/ 11/ 3/ 28
  IN2014/ 12/ 11/ 3/ 36

Lack!
COMMAND LINE_:2014/ 12/ 11/ 3/ 44
```

잔액이 500원인 경우 입장 실패!

DEMO

```
abc I2014/ 12/ 11/ 1/ 20
N
FEE : 1050
BALANCE : 700
Cardfile Write !
Readefile Write!
COMMAND LINE_:2014/ 12/ 11/ 1/ 28
ab2014/ 12/ 11/ 1/ 36
c OUT2014/ 12/ 11/ 1/ 44

FEE : 0
BALANCE : 700
Cardfile Write !
Readefile Write!
COMMAND LINE_:2014/ 12/ 11/ 1/ 52
```

```
abc2014/ 12/ 11/ 2/ 0
IN
Cardfile Write !
Readefile Write!
FEE : 0
BALANCE : 700
COMMAND LINE_:2014/ 12/ 11/ 2/ 8
2014/ 12/ 11/ 2/ 16
```

잔액이 700원인경우 버스에서 지하철 환승 성공

DEMO

```
abc IN
FEE : 1050
UALANCE : 47900
Cardfile Write !
Readefile Write!
COMMAND LINE_:2014/ 12/ 11/ 17/ 20
abc2014/ 12/ 11/ 17/ 28
OUT
FEE : 0
UALANCE : 47900
Cardfile Write !
Readefile Write!
COMMAND LINE_:2014/ 12/ 11/ 17/ 36
```

```
abc IN
Cardfile Write !
Readefile Write!
FEE : 0
UALANCE : 47900
COMMAND LINE_:2014/ 12/ 11/ 18/ 56
2014/ 12/ 11/ 19/ 4
2014/ 12/ 11/ 19/ 12
```

CA. C:₩WINDO

```
abc 2014/ 12/ 11/ 19/ 12
OUT
Cardfile Write !
Readefile Write!
FEE : 300
UALANCE : 47600
COMMAND LINE_:2014/ 12/ 11/ 19/ 20
```

한 정거장 지난 후 요금 300원 부과

DEMO

```
C:\#cygwin64\home\KimYongHyun\ExactCalclater>exactsystem.exe
===== EXACT CALCULATION RESULT =====
==== BUS : 1050 =====
==== METRO : 300 =====
===== EXACT CALCULATION RESULT =====
==== BUS : 0 =====
==== METRO : 0 =====
```

정산 후 BUS FILE 초기화 확인

IMPRESSION



IMPRESSION

김용현

팀으로 프로젝트를 하는 데에서 느낀 점이 크다. 좋은 팀을 만난 것 같다. 서로 의사소통과 역할분담 부분에서 서로 디버그도 도와주면서 진행한 덕에 조금 더딘 부분도 있었지만, 소프트웨어 공학에서 가장 중요하게 생각하는 Win-Win 전략에 4학기 중 가장 좋았던 부분이었다. Water Fall 모델에서 각 단계별 중요성을 뼈저리게 느꼈고, 자료구조, Thread, 알고리즘을 코딩 할 때 넣었다는 점에서 좋았다. 기승전술이 좋기도 했다.

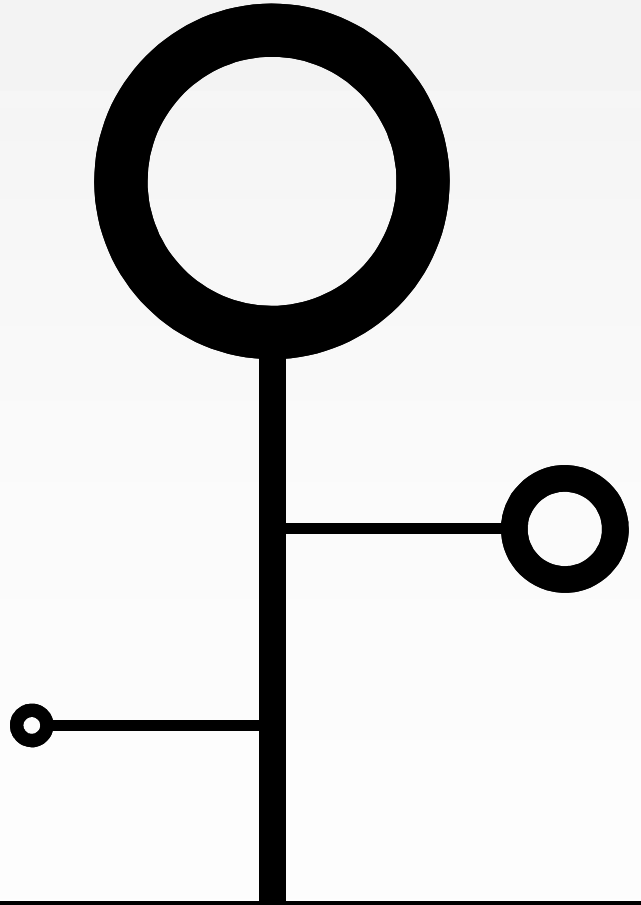
김준호

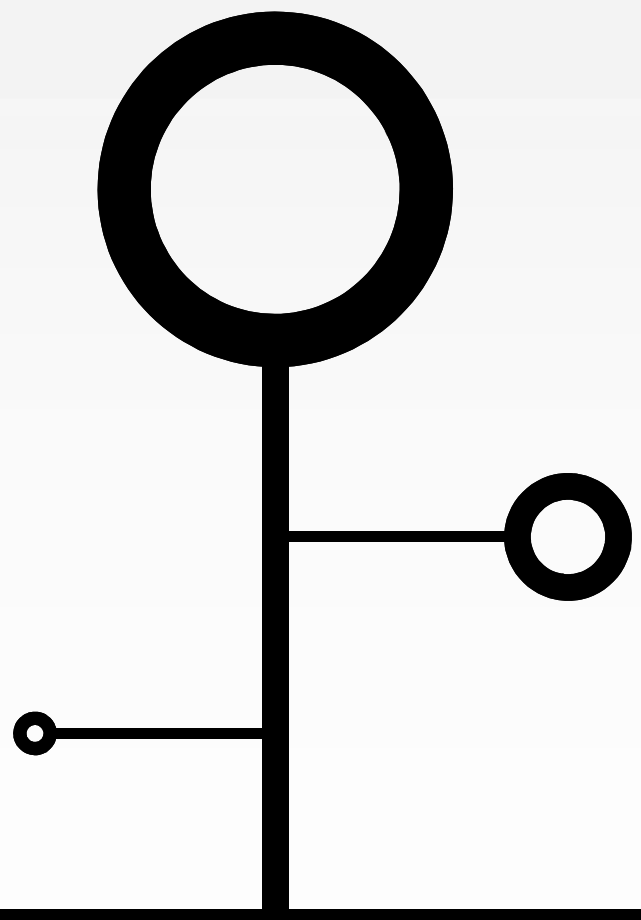
소공을 하며 가장 크게 느낀 점은 첫 설계가 무지막지하게 중요하다는 것을 뼈저리게 느꼈다. 다음 단계로 넘어가 문제가 발견되면 다시 돌아가 수정을 반복 해야 하기 때문이다. 학기초 SASD 작성하면서 막막해 머리를 쥐어 뜯은 것이 엇그제 같은데 벌써 이렇게 소공과 함께 한 학기를 마무리 했다는 점에서 뿌듯하다. 기회가 된다면 다음 학기에 소모를 듣고 싶다!

박정민 :

한 학기 동안 많은 시간을 들여서 하나의 프로그램을 완성 하였는데, 여태까지 학교 다니면서 이렇게 구조적 분석을 하여 프로그램을 만든 적이 없어서 한 단계 한 단계 진행 할 때 마다 새로운 경험을 했고, 데드라인이 다가옴에 따라 밤을 새고 씻지도 못하고 했지만, 제출하고 다음날 발표 한 뒤 집에 가서 쉴 때는 너무 행복했다. 그리고 이렇게 고생한 만큼 얻은 것이 있어서 보람 찾던 것 같다.

Q&A





THANK YOU