

기술인증테스트 시스템&솔루션개발 실기형 문제지

- 형식정보 -

[2019년 #차]

사번	성명
유의 사항	<div>1. 공정한 평가를 위해 동료들 도와주는 행위, 보여주는 행위를 금지하고 있습니다.</div> <div>2. 부정행위 적발 시, 응시한 평가는 0점 처리됩니다.</div> <div>3. 본 시험지는 응시장 외부로 유출할 수 없으며, 시험 종료 후 감독관에게 제출해야 합니다.</div>

1. 버스 정보 시뮬레이션

개요	<p>버스 정보 시뮬레이션 구현을 통해</p> <p>요구사항 분석, 파일처리, 데이터관리, 외부 프로그램 호출, 통신처리, 병렬 처리 등의 기술역량 및 프로그램 구현 역량을 측정하기 위한 문제입니다.</p>
폴더 정보	<div><p>※ 프로그램 및 파일 위치 정보 (실행위치 기반 상대경로 사용 필수)</p><ul style="list-style-type: none">- 구현할 프로그램 위치 및 실행 위치 : 각 문항 홈 (SUB1 / SUB2 / SUB3 / SUB4 / SUB5)- 입력 파일 위치 : 각 문항 홈 아래 <u>INFILE</u> 폴더 (LOCATION.TXT, STATION.TXT)- 출력 파일 위치 : 각 문항 홈 아래 <u>OUTFILE</u> 폴더 (PREPOST.TXT, ARRIVAL.TXT)- 제공되는 통신 클라이언트 프로그램 위치 및 파일명 : 각 문항 홈 아래 CLIENT.EXE- 제공되는 정류장도착예정시각출력 프로그램 위치 및 파일명 : 각 문항 홈 아래 SINAGE.EXE- 자가 검수용 참고 파일명 : OUTFILE폴더 내 CMP_*.TXT<p>* 제공되는 파일들은 문항에 따라 다를 수 있음</p></div>
실행 방식	<div><p>※ 구현할 프로그램 형식</p><ul style="list-style-type: none">- 프로그램 형태 : 콘솔(Console) 프로그램- 프로그램 파일명 : SP_TEST- 실행 방식(문항1~3) : 콘솔 실행→결과처리→자동종료<div><p>C:\W>SP_TEST<엔터키> ← 콘솔 실행 / 결과처리 후 프로그램 자동 종료</p><p>C:\W></p></div><ul style="list-style-type: none">- 실행방식(문항4~5) : 콘솔 실행→통신연결대기→통신처리 (종료 없음)<div><p>C:\W>SP_TEST<엔터키></p></div></div>

형식 정보

1. "위치정보파일" 형식

- 파일명 : LOCATION.TXT (INFILE 폴더 내)
- 데이터 포맷 (문항 1) : 시각(HH:MM:SS)
 - + "#BUS01," + BUS01의 기점으로부터의 거리(5자리 숫자, 단위:m)
 - + "#BUS02," + BUS02의 기점으로부터의 거리(5자리 숫자, 단위:m)
 - + "#BUS03," + BUS03의 기점으로부터의 거리(5자리 숫자, 단위:m)
- 데이터 포맷 (문항 2,3) : 시각(HH:MM:SS)
 - + "#" + 차량번호(BUS01~BUS99) + "," + 기점으로부터의 거리(5자리 숫자, 단위:m)
 - (1대~99대의 차량 대수만큼 반복)
- 위치정보 손실 발생시 (문항 3) : 시각(HH:MM:SS)
- 최종출력커맨드 : "PRINT"

10:00:00#BUS01,02345#BUS02,00800#BUS03,07843
10:00:01#BUS01,02362#BUS02,00805#BUS03,07847
10:00:02#BUS01,02378#BUS02,00810#BUS03,07850
10:00:03#BUS01,02395#BUS02,00815#BUS03,07854
PRINT

← 문항(1)
3대의 차량만 존재

← 출력 커맨드 (10:00:03 시점)

10:00:00#BUS01,02345#BUS02,00800#BUS03,07843#BUS04,08504
10:00:01#BUS01,02362#BUS02,00805#BUS03,07847#BUS04,08508
10:00:02#BUS01,02378#BUS02,00810#BUS03,07850#BUS04,08511
10:00:03#BUS01,02395#BUS02,00815#BUS03,07854#BUS04,08515
PRINT

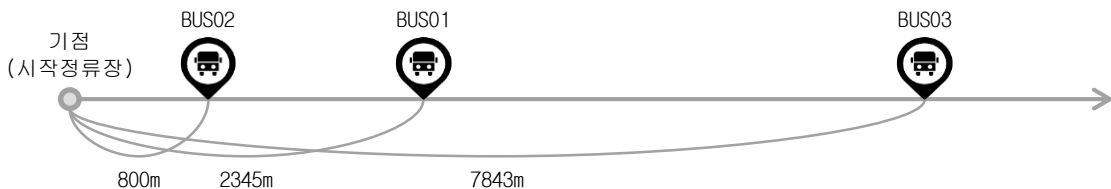
← 문항(2)
N대의 차량 존재

...
10:59:58#BUS01,02150#BUS02,00855#BUS03,07500
10:59:59#BUS01,02160#BUS02,00870#BUS03,07515
11:00:00#BUS01,02170#BUS02,00885#BUS03,07530
11:00:01#BUS01,02180#BUS02,00900#BUS03,07545
11:00:02#BUS01,02190#BUS02,00920#BUS03,07560
11:00:03#BUS01,02200#BUS02,00940#BUS03,07570
11:00:04#BUS01,02210#BUS02,00960#BUS03,07580
11:00:05
11:00:06 } 11:00:05~11:00:06 시점의 위치정보 손실
PRINT

← 문항(3)
N대의 차량 존재
위치정보 손실 발생

- 위치정보 손실이 발생하기 시작하면 지속 발생하는 것으로 가정.

- (참고) 문항(1) 10:00:00 시점의 위치정보



형식 정보

2. “정류장정보파일” 형식

- 파일명 : STATION.TXT (INFILE 폴더 내)
- 데이터 포맷 (문항2) : 정류장번호(STA01~STA99)
+ “#” + 기점으로부터의 거리(5자리 숫자, 단위:m)
- 데이터 포맷 (문항3, 4, 5) : 정류장번호(STA01~STA99)
+ “#” + 기점으로부터의 거리(5자리 숫자, 단위:m)
+ “#” + 정류장 구간 제한 속도(3자리 숫자, 단위:km/h)

STA01#00000
STA02#02375
STA03#04750
...

← 기점으로부터의 거리가 2375m

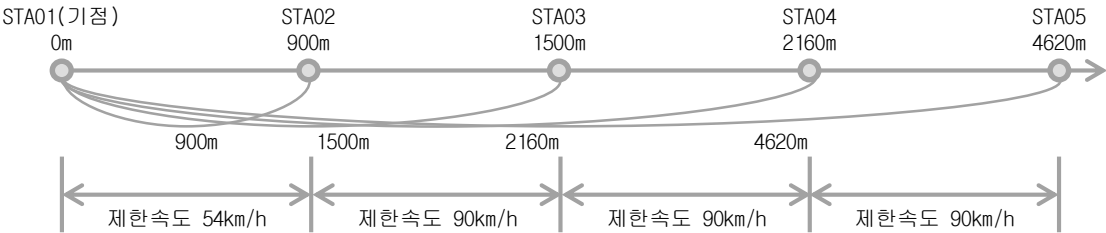
← 문항(2)

STA01#00000#054
STA02#00900#090
STA03#01500#090
STA04#02160#090
STA05#04620#072
...

← STA01~STA02 구간 제한 속도 54km/h

← 문항(3,4,5)

- (참고) 정류장 정보



형식 정보

3. “선/후행차량정보파일” 형식

- 파일명 : PREPOST.TXT (OUTFILE 폴더 내)
- 데이터 포맷 : 시각(HH:MM:SS)
 - + “#” + 차량번호(BUS01~BUS99)
 - + “#” + 선행 차량번호(BUS01~BUS99, 미존재시 “NOBUS”)
 - + “,” + 선행 차량과의 거리(5자리 숫자, 단위:m, 미존재시 “00000”)
 - + “#” + 후행 차량번호(BUS01~BUS99, 미존재시 “NOBUS”)
 - + “,” + 후행 차량과의 거리(5자리 숫자, 단위:m, 미존재시 “00000”)
- (주의) 차량번호 오름차순으로 정렬

```
10:00:03#BUS01#BUS03,05459#BUS02,01580
10:00:03#BUS02#BUS01,01580#NOBUS,00000
10:00:03#BUS03#NOBUS,00000#BUS01,05459
```

4. “정류장도착예정정보파일” 형식

- 파일명 : ARRIVAL.TXT (OUTFILE 폴더 내)
- 데이터 포맷 : 시각(HH:MM:SS)
 - + “#” + 정류장번호(STA01~STA99)
 - + “#” + 정류장으로부터 가장 가까운 도착 예정 차량번호(BUS01~BUS99, 미존재시 “NOBUS”)
 - + “#” + 정류장으로부터의 거리 (5자리 숫자, 단위:m, 미존재시 “00000”)
- (주의) 정류장번호 오름차순으로 정렬

```
10:00:03#STA01#NOBUS,00000
10:00:03#STA02#BUS02,01560
10:00:03#STA03#BUS01,02355
...
```

형식 정보

5. 위치정보 손실을 고려한 최종 위치 계산

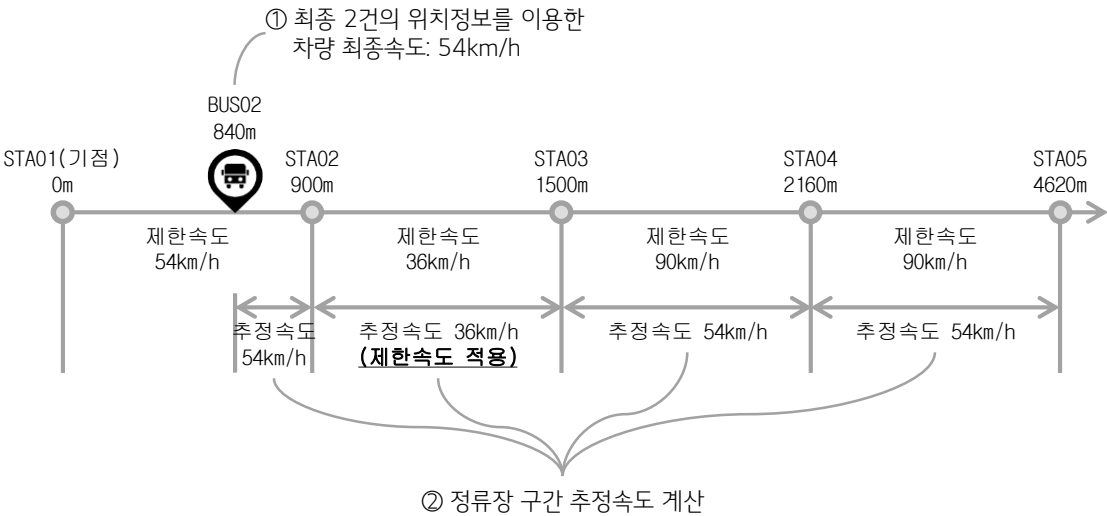
- ① 위치정보 손실 전 최종 2건의 위치정보를 이용하여 차량의 최종속도 계산
- ② 차량은 최종속도로 계속 운행한다고 가정하고 정류장 구간 제한속도를 초과할 수 없음을 고려하여 정류장 구간 추정속도 계산
- 예> ①에서 계산된 차량의 최종속도는 54km/h이지만, STA02~STA03 구간의 제한속도가 36km/h 이므로 추정속도는 36km/h
- ③ ②에서 계산된 추정속도로 최종위치 계산
- (주의) 소수점 계산 시 최종 값 도출 후 버림 처리 후 출력

10:59:56#BUS02,00795
10:59:57#BUS02,00810
10:59:58#BUS02,00825
10:59:59#BUS02,00840
11:00:00
11:00:01
...
11:00:04
11:00:05
PRINT

① 최종 2건의 위치정보를 이용한
차량 최종속도 계산 : 54km/h

위치정보 손실

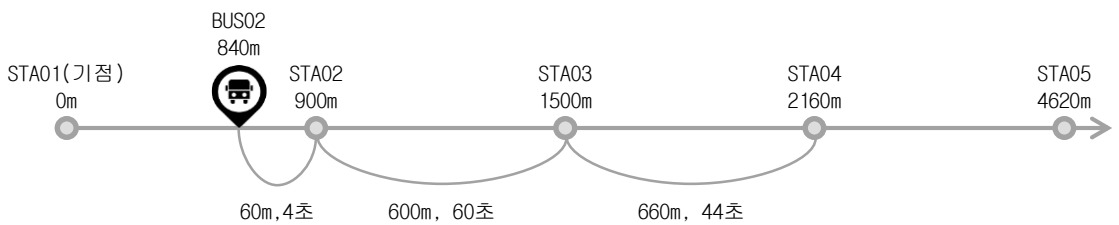
→ ③ 11:00:05 시점의 최종위치 계산 : 920m



형식 정보

6. 정류장도착예정시각 계산

- 위치정보 데이터를 이용하여 정류장 구간 추정속도를 계산 (5page 형식5 참고)
- 도착 예정 정류장과와의 거리를 정류장 구간으로 나누고, 정류장 구간 추정속도를 적용하여 도착 예정 시각 계산
- 예를 들어, BUS02 차량의 10:59:59 시점의 최종위치가 840m인 경우 STA04 정류장의 도착예정시각은 108초 후인 11:01:47 임
- (주의) 소수점 계산 시 최종 값 도출 후 버림 처리 후 출력



구간	추정속도	거리	소요시간
BUS02~STA02	54km/h	60m	4초
STA02~STA03	36km/h (제한속도 적용)	600m	60초
STA03~STA04	54km/h	660m	44초
합계		1320m	108초

형식 정보

7. 제공프로그램 형식 (정류장도착예정시각출력 프로그램)

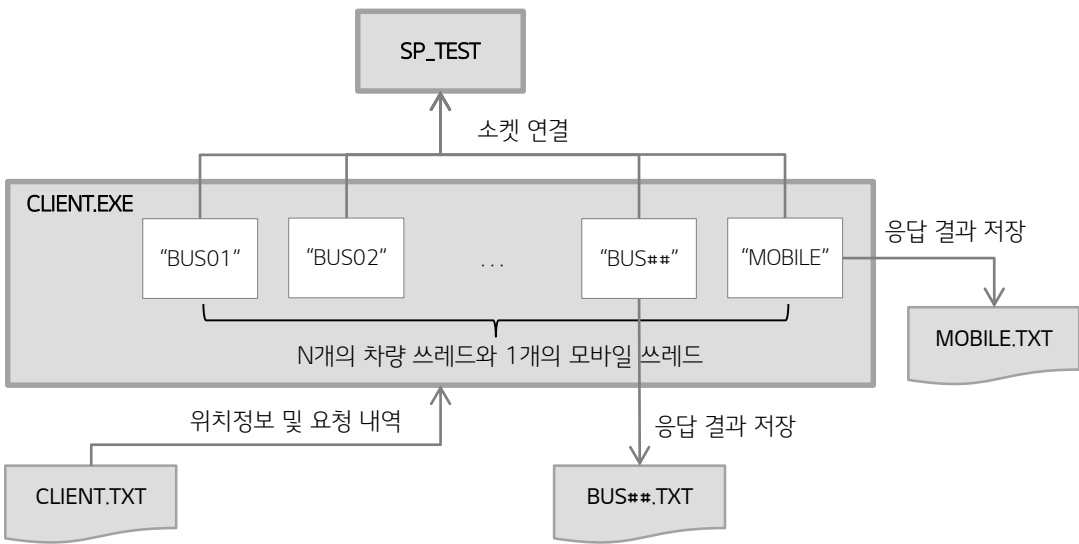
- 프로그램 형태 : 콘솔(Console) 프로그램
- 프로그램 파일명 : SIGNAGE.EXE
- 실행 방식(콘솔실행) : SIGNAGE.EXE를 실행하여 정류장도착예정시각 정보를 콘솔에 입력 (Standard Input으로 정류장도착예정시각 정보 전달)
- 데이터 포맷 : 시각(HH:MM:SS)
 - + “#” + 정류장번호(STA01~STA99)
 - + “#” + 가장 가까운 도착 예정 차량번호(BUS01~BUS99, 미존재시 “NOBUS”)
 - + “#” + 도착 예정 시각 (HH:MM:SS, 미존재시 “00:00:00”)
- (참고) SIGNAGE.EXE에 전달된 정류장도착예정시각은 OUTFILE 폴더 내에 SIGNAGE.TXT 파일로 저장됨
- 아래는 이해를 돕기 위한 콘솔에서 단독으로 실행한 예시이며,
이를 참고하여 구현하는 프로그램에서는 외부 프로그램 호출 방식을 적용해야 함

```
C:\W>SIGNAGE.EXE<엔터키>
11:00:06#STA01#NOBUS,00:00:00<엔터키>
11:00:06#STA02#NOBUS,00:00:00<엔터키>
11:00:06#STA03#BUS02,11:00:31<엔터키>
11:00:06#STA04#BUS02,11:01:04<엔터키>
11:00:06#STA05#BUS02,11:03:07<엔터키>
11:00:06#STA06#BUS02,11:04:34<엔터키>
11:00:06#STA07#BUS02,11:05:54<엔터키>
11:00:06#STA08#BUS03,11:01:56<엔터키>
```


형식 정보

8. 제공프로그램 형식 (통신 클라이언트 프로그램)

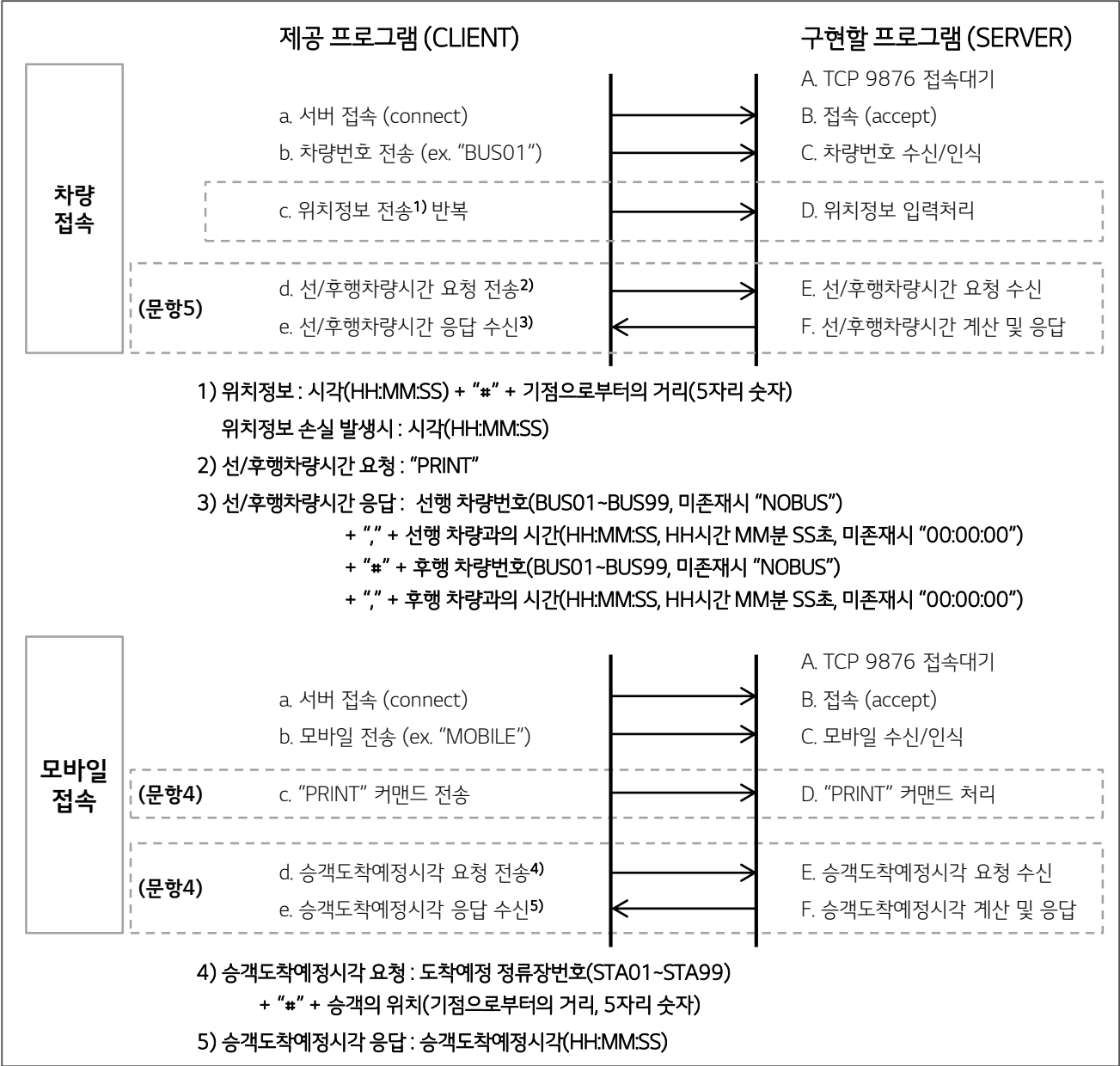
- 프로그램 형태 : 콘솔(Console) 프로그램
- 프로그램 파일명 : CLIENT.EXE
- 실행 방식(콘솔실행) : CLIENT
- N개의 차량 쓰레드와 1개의 모바일 쓰레드가 생성되어 SP_TEST 서버 프로그램으로 소켓 연결
- (참고) 통신 클라이언트 프로그램(CLIENT.EXE)은 INFILE 폴더 내 CLIENT.TXT 파일을 읽어 위치 정보 송신 및 요청을 수행함 (자가 검수를 위한 테스트시 참고)
- (참고) 통신 클라이언트 프로그램(CLIENT.EXE)이 수신한 승객도착예정시각 응답 결과는 OUTFILE 폴더 내에 MOBILE.TXT 파일로 저장됨 (문항4)
- (참고) 통신 클라이언트 프로그램(CLIENT.EXE)이 수신한 선/후행차량시간 응답 결과는 OUTFILE 폴더 내에 BUS##.TXT 파일로 저장됨 (문항5)



형식 정보

9. 통신 절차 형식 (통신 클라이언트 프로그램) (1/2)

- 여러 소켓으로부터 동시 접속요청 처리해야 함 (N개의 차량 접속 및 1개의 모바일 접속)
- 접속 상태를 유지하며 차량 접속으로부터 지속적으로 위치정보를 수신
- (문항4) 모바일 접속으로부터 "PRINT" 커맨드 수신시 다음의 기능 수행
 - 1) 선/후행차량정보파일 기록
 - 2) 정류장도착예정정보파일 기록
 - 3) 정류장도착예정시간 계산 및 외부프로그램 호출문항3에서 이미 구현한 기능
- (문항4) 모바일 접속으로부터 승객도착예정시간 요청 수신시 승객도착예정시간 계산하여 응답 수행
- (문항5) 차량 접속으로부터 선/후행차량시간 요청 수신시 선/후행차량시간 계산하여 응답 수행



※ 모든 송수신 데이터의 끝은 LF(0x0A, '\n')로 종료됨.

형식 정보

9. 통신 절차 형식 (통신 클라이언트 프로그램) (2/2)

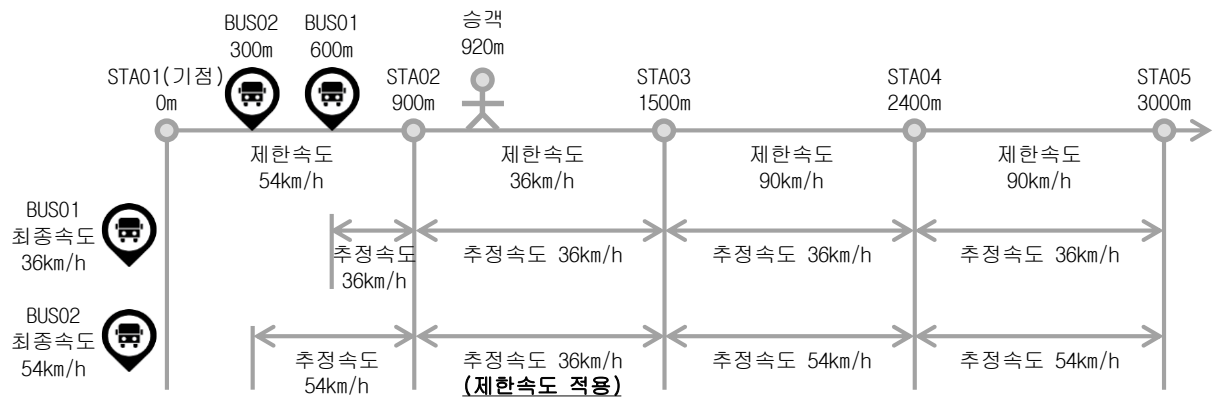
- (참고) N개의 차량 접속 및 1개의 모바일 접속의 데이터 송수신 시나리오

시각 (초)	BUS01	BUS02	BUS03	MOBILE	SERVER 처리
0	소켓 connect	소켓 connect	소켓 connect	소켓 connect	N개의 차량과 1개의 모바일 접속
1	BUS01	BUS02	BUS03	MOBILE	차량/모바일 인식
2	10:59:56#02130	10:59:56#00825	10:59:56#07470		위치정보 입력 처리
3	10:59:57#02140	10:59:57#00840	10:59:57#07485		위치정보 입력 처리
4	10:59:58#02150	10:59:58#00855	10:59:58#07500		위치정보 입력 처리
5	10:59:59#02160	10:59:59#00870	10:59:59#07515		위치정보 입력 처리
6	11:00:00#02170	11:00:00#00885	11:00:00#07530		위치정보 입력 처리
7	11:00:01#02180	11:00:01#00900	11:00:01#07545		위치정보 입력 처리
8	11:00:02#02190	11:00:02#00920	11:00:02#07560		위치정보 입력 처리
9	11:00:03	11:00:03#00940	11:00:03#07570		위치정보 입력 처리 (일부 손실)
10	11:00:04	11:00:04	11:00:04#07580		위치정보 입력 처리 (일부 손실)
11	11:00:05	11:00:05	11:00:05		위치정보 입력 처리 (전체 손실)
12	11:00:06	11:00:06	11:00:06		위치정보 입력 처리 (전체 손실)
13				PRINT	(문항4) 1) 선/후행차량정보파일 기록 2) 정류장도착예정정보파일 기록 3) 정류장도착예정시간 계산 및 외부프로그램 호출
14				요청> STA07#02130 응답> 11:05:54	(문항4) 승객도착예정시간 계산 및 응답
15			요청> PRINT 응답> NOBUS#00:00:00# BUS02#00:05:52		(문항5) 선/후행차량시간 계산 및 응답

형식 정보

10. 승객도착예정시각 계산

- 도보 이동 속도가 3.6km/h(1m/s)인 승객이 인접 정류장으로 이동 후 버스를 1회 이용했을 때 요청 정류장의 가장 빠른 도착예정시각 계산 (이용할 버스가 없거나 도보가 더 빠른 경우 대상 정류장까지 도보로 이동)
- 예를 들어, 10:00:00 시점에 BUS01(600m)과 BUS02(300m)가 운행중인 상황에서 920m에 위치한 승객이 STA04까지 이동하고자 할 때 가장 빠른 도착예정시각은 10:02:40임
- (주의) 소수점 계산 시 최종 값 도출 후 버림 처리 후 출력



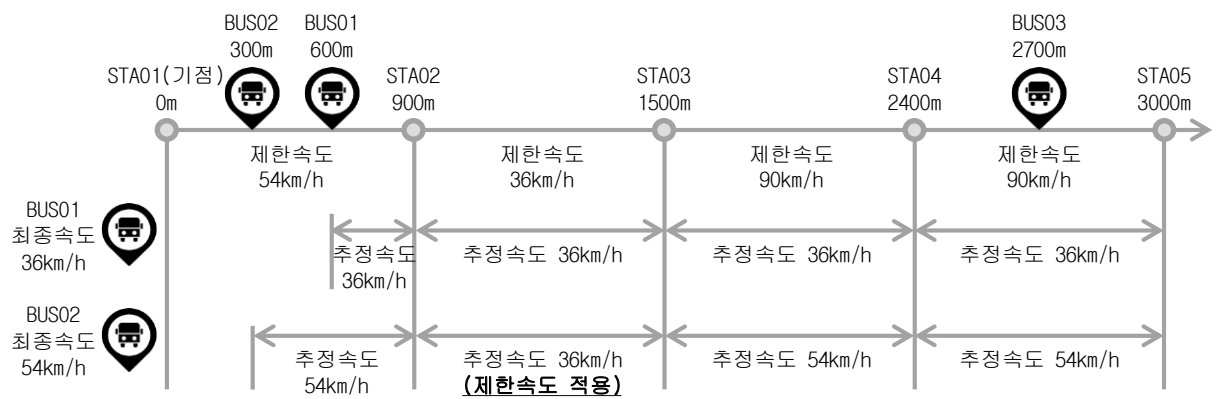
	출발시각	STA02 도착예정시각	STA03 도착예정시각	STA04 도착예정시각
BUS01	10:00:00	10:00:30	10:01:30	10:03:00
BUS02	10:00:00	10:00:40	10:01:40	10:02:40
승객	10:00:00	10:00:20 (도보) 10:00:40 (BUS02 승차 ¹⁾)	10:01:40 (BUS02)	10:02:40 (BUS02)

1) STA02 정류장에 BUS01 차량이 먼저 도착하지만 승차하지 않고 BUS02 차량에 승차 (BUS02 차량이 STA04 정류장에 먼저 도착하므로)

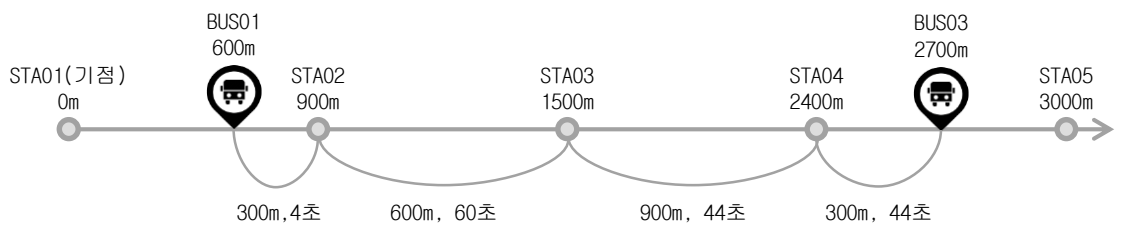
형식 정보

11. 선/후행차량시간 계산 (1/2)

- 위치정보 데이터를 이용하여 정류장 구간 추정속도를 계산 (5page 형식5 참고)
- 선/후행차량과의 거리를 정류장별 구간으로 나누고, 정류장 구간 추정속도를 적용하여 시간 계산
- (주의 1) 선행차량과의 시간은 기준차량의 추정속도로, 후행차량과의 시간은 후행차량의 추정속도로 계산
- 예를 들어, BUS01 차량의 선/후행차량시간을 계산하는 경우

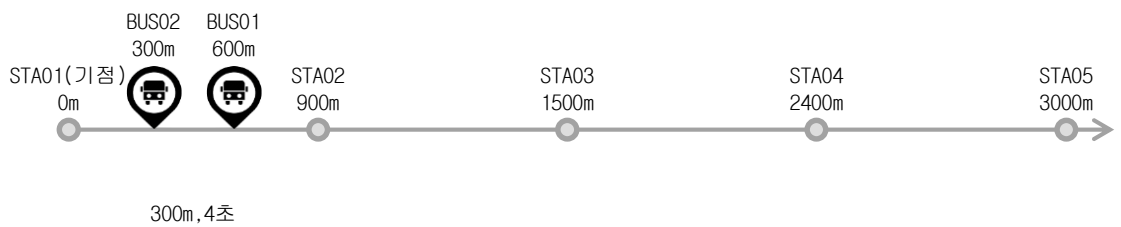


- 선행차량과의 시간은 기준차량(BUS01)의 속도로 계산



구간	추정속도	거리	소요시간
BUS01-STA02	36km/h	300m	30초
STA02-STA03	36km/h	600m	60초
STA03-STA04	36km/h	900m	90초
STA04-BUS03	36km/h	300m	30초
합계		2100m	210초 (00:03:30)

- 후행차량과의 시간은 후행차량(BUS02)의 속도로 계산

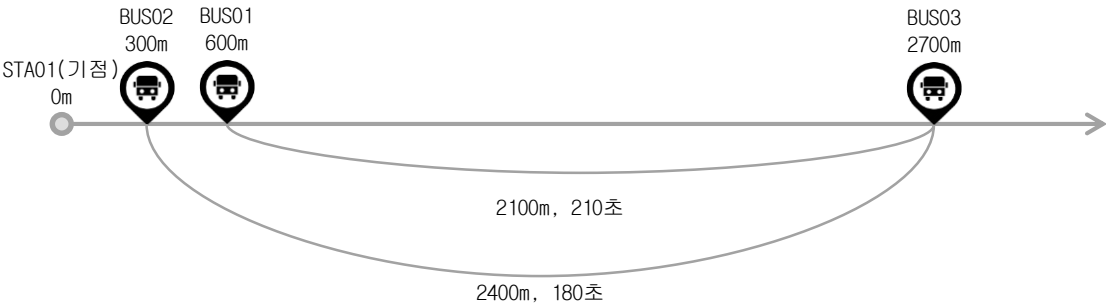


구간	추정속도	거리	소요시간
BUS02-BUS01	54km/h	300m	20초
합계		300m	20초 (00:00:20)

형식 정보

11. 선/후행차량시간 계산 (2/2)

- (주의 2) 차량간 거리가 아닌 시간을 기준으로 선/후행차량의 찾을
- 예를 들어, BUS03 차량의 후행차량을 찾을 때 거리 기준으로 BUS01이 더 근접하지만, 시간 기준으로 BUS02와의 시간이 더 적으므로 후행차량을 BUS02로 판단



- (주의 3) 소수점 계산 시 최종 값 도출 후 버림 처리 후 출력

