

기술인증테스트 시스템&솔루션개발 실기형 문제지

[2020년 #차]

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

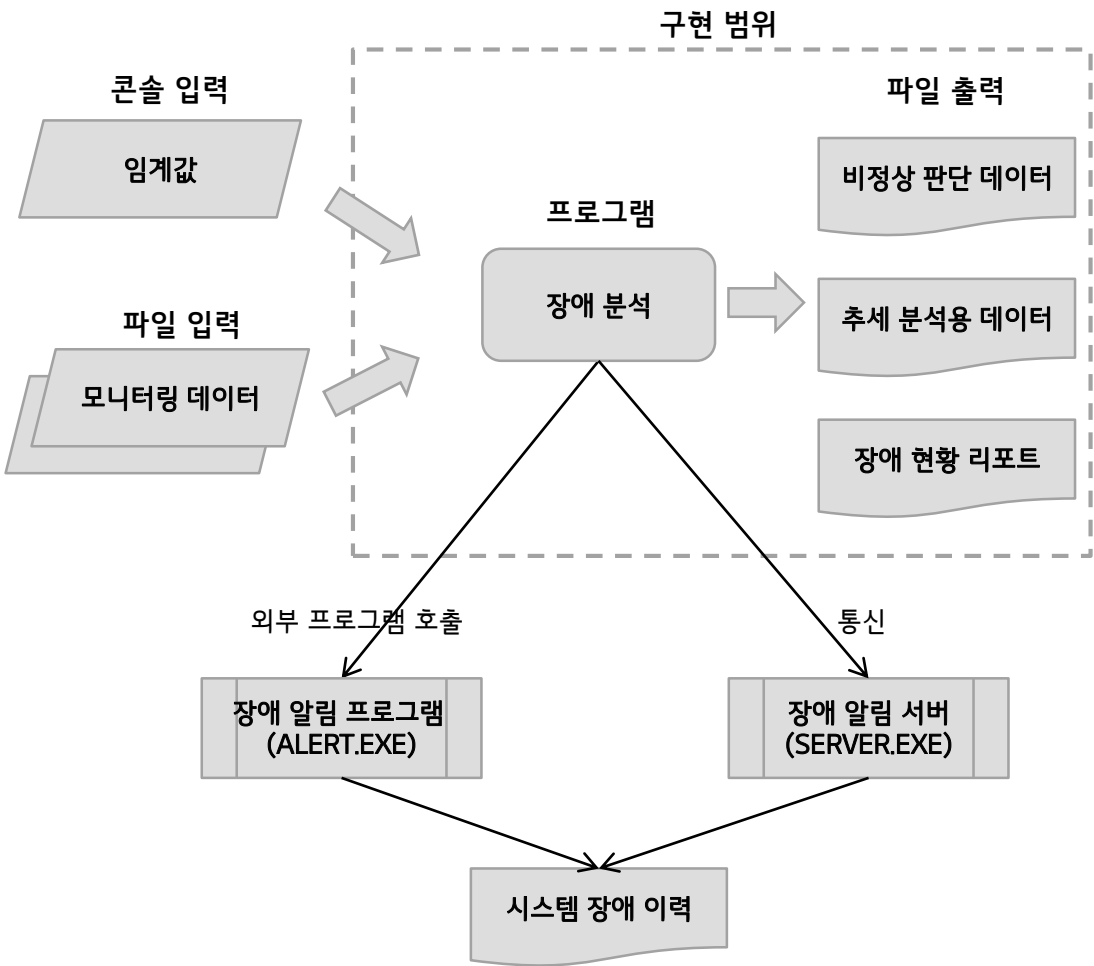
장애 분석 시스템

개요

해당 시스템 구현을 통해 요구사항 분석, 파일처리, 데이터관리, 외부 프로그램 호출, 통신처리 등의 기술역량 및 프로그램 구현 역량을 측정하기 위한 문제입니다.

설명

본 프로그램은 사용자가 입력한 임계값을 기준으로 모니터링 데이터에서 초과 여부를 판별하여 비정상 판단 데이터를 생성하고, 이를 추세 분석용 데이터로 변환하고 시스템 장애 여부를 분석하여 장애 알림 프로그램으로 전달합니다. 그리고 시스템 장애 판별 시 장애 원인 프로세스를 검출하여 장애 알림 서버로 전달하고 장애 현황 리포트를 생성하는 장애 분석 프로그램입니다.



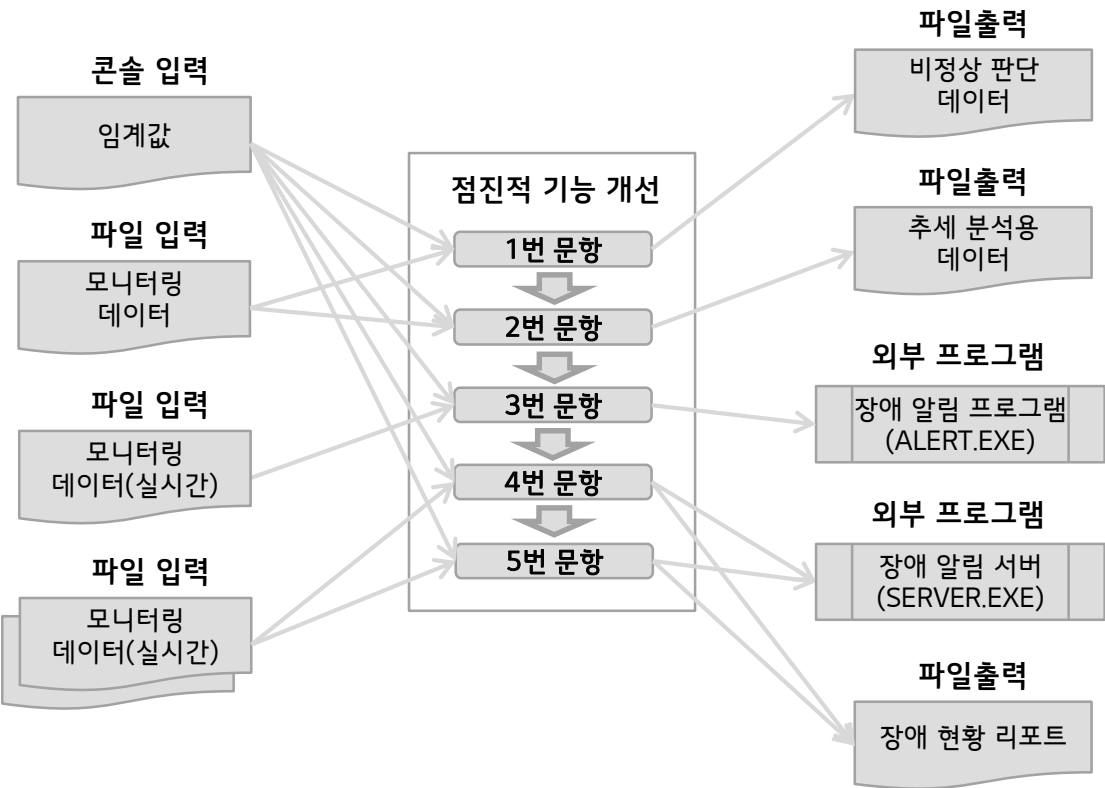
[기능 요약]

- 사용자가 콘솔로 입력한 임계값을 기준으로 모니터링 데이터에서
 - 임계값 초과 여부를 판별하여 비정상 판단 데이터를 생성하고, 추세 분석용 데이터로 변환한다
- 사용자가 콘솔로 입력한 임계값을 기준으로 실시간으로 수집되는 모니터링 데이터에서
 - 임계값 초과 여부를 판별하여 비정상 판단 데이터를 생성하고, 추세 분석용 데이터로 변환한다
 - 시스템 장애 여부를 분석하여 생성된 데이터를 장애 알림 프로그램으로 전달한다
 - 시스템 장애 판별 시, 장애 원인 프로세스를 검출하여 장애 알림 서버로 전달한다
 - 장애 알림 결과 응답에 따라 장애 현황 리포트를 생성한다

주의사항

- 실행 결과로 평가하고 부분점수는 없으므로 아래사항을 필히 주의해야 함
- 구현된 프로그램은 실행 완결성 필수 (명확한 실행&종료 처리, 정확한 결과 출력, 통상의 실행 시간)
 - 소 문항별 결과 검수 필수 (선행문항 오류 시, 후속문항 전체에 오류가 발생할 수 있음)
 - 제시된 조건이 없는 한 선행요구사항 유지 필수 (소 문항별 입출력 관계도 참고)
 - 프로그램 실행 위치 및 실행결과출력 (위치, 파일명, 데이터포맷)은 요구사항과 정확히 일치 필수
 - 제시된 모든 위치는 상대경로 사용 필수 (프로그램 실행 위치 기준)
 - 종료조건에 맞는 자동종료 처리 필수 (불필요한 종료방해처리(pause/입력대기 등)를 하면 안됨)
 - 모든 문자는 대문자 처리 필수

문항 관계



문제

아래 제시된 문항은 문항번호가 증가할 수록 점진적 개선을 요구하는 방식으로 구성되어 있으며, 제시된 문항번호 별로 각각 구현된 소스와 컴파일 된 실행파일을 제출하시오.
cf) 1번 구현 → 1번 소스복사 → 2번 구현 → 2번 소스복사 → ...

1. 콘솔로 입력된 '임계값'을 기준으로, '모니터링 데이터'에서 시스템자원사용률이 임계값을 초과하는지 확인하여 비정상 여부인지 판단한 후 'Y' 또는 'N'을 추가하여 파일로 출력하시오.(20점)
- 파일출력 : 읽은 순서대로 '비정상 판단 데이터' 파일에 출력

상세설명

※ 임계값 콘솔 입력

- 입력 포맷 : 임계값(3, 범위 : 000 ~ 100)

- 입력 예시

C:\>SP_TEST<엔터키> ← 구현한 프로그램 실행 (Argument 없음)

070<엔터키> ← 콘솔 입력(임계값 70을 입력)

C:\>

※ 모니터링 데이터 형식 정보

- 파일명 : MONITORING.TXT (INPUT 폴더 내)

- 데이터 포맷 :

5개 프로세스의
자원 사용률
(프로세스 개수 고정)

순번(4) + "#" + "SYS" + ":" + 시스템자원사용률(3) + "#" +
프로세스이름(6) + ":" + 프로세스자원사용률(3) + "#" +
.....
프로세스이름(6) + ":" + 프로세스자원사용률(3)

- 입력 데이터 파일 예시 (임계값 : 70)

0001#SYS:086#PROC01:012#PROC02:030#PROC03:024#...#PROC05:005

0002#SYS:034#PROC01:015#PROC02:012#PROC03:002#...#PROC05:003

0003#SYS:095#PROC01:020#PROC02:001#PROC03:012#...#PROC05:017

...

초과

초과

5개 프로세스

'시스템자원사용률'

'프로세스자원사용률'

Y or N을 추가함

※ 비정상 판단 데이터 파일 형식 정보

- 파일명 : REPORT.TXT (OUTPUT 폴더 내)

- 데이터 포맷 : '모니터링 데이터' 포맷 + "#" + 비정상 여부(1, 비정상:'Y' 또는 정상:'N')

- 출력 파일 예시

0001#SYS:086#PROC01:012#PROC02:030#PROC03:024#...#PROC05:005#Y

0002#SYS:034#PROC01:015#PROC02:012#PROC03:002#...#PROC05:003#N

0003#SYS:095#PROC01:020#PROC02:001#PROC03:012#...#PROC05:017#Y

...

- 문제
2. 위 1번 문항까지 구현된 내용을 기준으로, 추세 분석을 위해 '비정상 판단 데이터'에서 시스템자원사용률의 **이동 평균값을 계산**하고 그 값을 추가하여 파일로 출력하시오. (15점)

- 이동 평균값 계산 처리 추가 : '※이동 평균값 계산 규칙' 참조

- 파일출력 변경 : '추세 분석용 데이터'를 파일에 출력

상세설명

※ 이동 평균값 계산 규칙

- 현재 데이터와 이전 2개 데이터(**총 3개**)의 평균값을 구함(소수점 이하 버림)

- 현재 데이터와 이전 데이터의 수가 **3개 미만**인 경우, **해당 개수 만큼**의 평균값을 구함

- 규칙 예시 :

1번 문항에서 구한 '비정상 판단 데이터'를 이용

0001#SYS:086#PROC01:012#...#PROC05:005#Y ← [총 1개 평균값] **86** = (86) / 1

0002#SYS:034#PROC01:015#...#PROC05:003#N ← [총 2개 평균값] **60** = (86+34) / 2

0003#SYS:095#PROC01:020#...#PROC05:017#Y ← [총 3개 평균값] **71** = (86+34+95) / 3

0004#SYS:087#PROC01:024#...#PROC05:004#Y ← [총 3개 평균값] **72** = (34+95+87) / 3

... '시스템자원사용률'

※ 추세 분석용 데이터 파일 형식 정보

- 파일명 : REPORT.TXT (OUTPUT 폴더 내)

- 데이터 포맷 : '비정상 판단 데이터' 포맷 + **"#" + 이동 평균값(3)**

- 출력 파일 예시

0001#SYS:086#PROC01:012#...#PROC05:005#Y**#086**

0002#SYS:034#PROC01:015#...#PROC05:003#N**#060**

0003#SYS:095#PROC01:020#...#PROC05:017#Y**#071**

0004#SYS:087#PROC01:024#...#PROC05:004#Y**#072**

...

'이동 평균값'을 추가함

평가대상

프로그램 정상 실행, 출력된 파일 내용(REPORT.TXT)

Secret

4

LG CNS

문제

3. 위 2번 문항까지 구현된 내용을 기준으로, '추세 분석용 데이터'에서 '시스템 장애 판별 규칙'에 따라 **장애를 판별**하고 이를 알리기 위해 '장애 알림 프로그램'을 호출하시오.(15점)

- '모니터링 데이터' 처리 방식 변경 : 생성 방식이 변경된 '모니터링 데이터(실시간)' 파일에 **실시간으로 추가**되는 데이터를 약 0.1초 주기로 확인하여 처리하도록 변경 (Polling 방식)
'※모니터링 데이터(실시간)의 생성 방식' 참조

- 시스템 장애 판별 기능 추가 : '※시스템 장애 여부 판별 규칙' 참조

- 장애 알림 프로그램 호출 기능 추가 : '※장애 알림 프로그램 호출 규칙' 참조

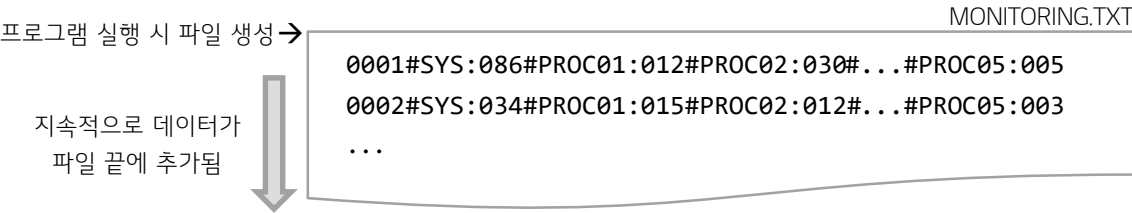
- 파일출력 변경 : '시스템 장애'로 판별된 '추세 분석용 데이터'를 파일로 직접 출력하지 않고,
'장애 알림 프로그램'을 호출하도록 변경

상세설명

- ※ 모니터링 데이터(실시간)의 생성 방식 : 사전 제공 -> 실시간 추가
- 제공 프로그램(CREATE_MONITORING.EXE)을 콘솔에서 실행하면, 신규로 MONITORING.TXT 파일이 생성되고 데이터가 지속적으로 파일 끝에 추가됨 (생성이 끝나면 자동 종료됨)
 - 콘솔에서 CREATE_MONITORING.EXE를 실행한 후, 구현 프로그램을 실행해야 함
 - 제공 프로그램 실행 예시 :

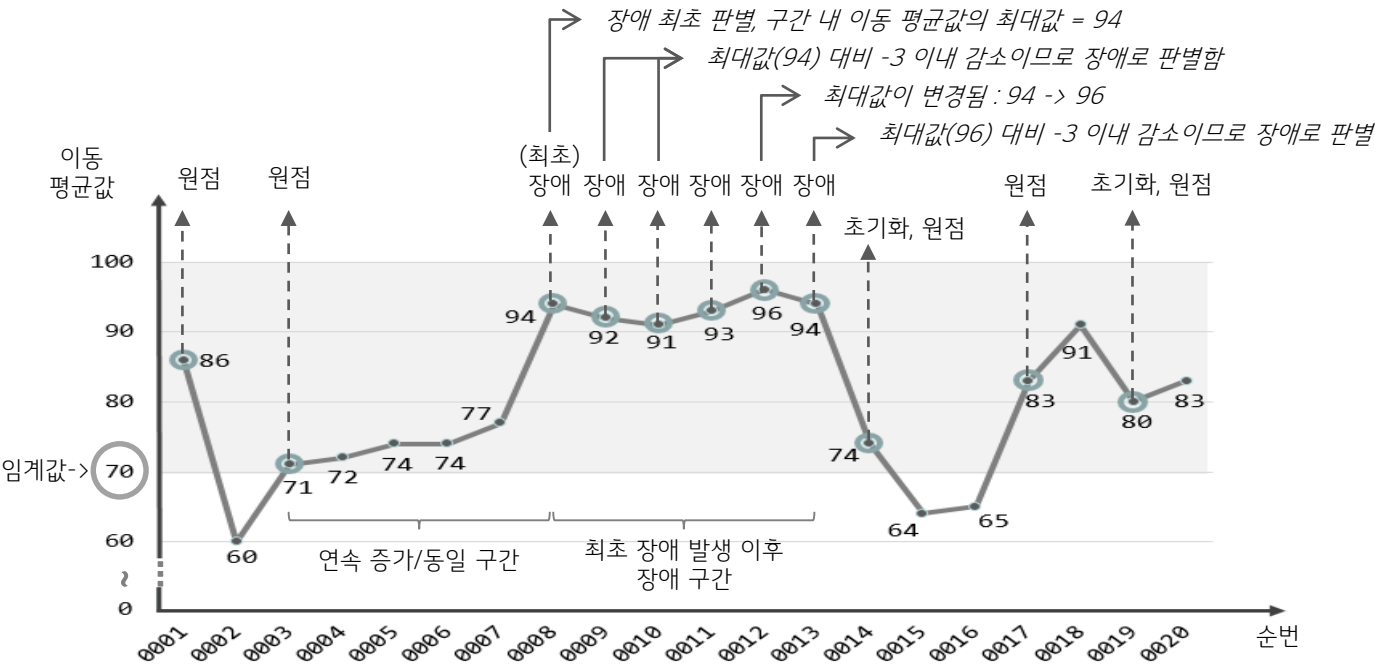
```
C:\>CREATE_MONITORING.EXE<엔터키>    ← SUPPORT 폴더에서 제공 프로그램 실행
...
SP_TEST 실행 후 엔터를 입력하세요
PAUSE<엔터키>                        ← 별도의 콘솔에서 SP_TEST 실행 후 엔터 입력
...
C:\>                                  ← 데이터 생성이 완료되면 자동 종료됨
```

- 데이터 포맷 : '모니터링 데이터' 포맷과 동일(1번 문항과 동일)
- 데이터 생성 예시 :



상세설명
(계속)

- ※ 시스템 장애 여부 판별 규칙
 - 원점(횡수 측정 시작 지점) 조건 :
 - > 이동 평균값이 임계값을 **최초로 초과**한 경우
 - > 이동 평균값이 감소하여 횡수 측정을 초기화 하지만, 여전히 이동 평균값이 임계값을 초과한 경우
 - 시스템 장애 조건 :
 - > '임계값'을 초과한 이동 평균값이 연속으로 **5회 이상 증가/동일**한 경우 시스템 장애로 최초 판별
 - > 시스템 장애 **최초 판별 이후**에는, 최초 판별 이후 이동 평균값의 최대 값보다 **-3이내 감소**도 횡수에 포함
 - > 원점은 증가/동일 횡수에 **포함하지 않음**
 - 규칙 예시 (임계값 : 70)



- ※ 장애 알림 프로그램 (외부 프로그램) 호출 규칙
 - '시스템 장애'가 판별되면, **프로그램 내부에서 '장애 알림 프로그램'(ALERT.EXE)을 호출**함
 - 호출 시 '추세 분석용 데이터'를 Argument로 전달함
 - 입력 Argument : '시스템 장애'로 판별된 '추세 분석용 데이터'

'입력 Argument'의 예

0008#SYS:091#PROC01:038#...#PROC05:009#Y#094

장애 알림 프로그램
(ALERT.EXE)

파일 출력
REPORT.TXT

'장애 알림 프로그램'은 전달받은 데이터를 가공 없이 자동으로 파일로 출력함

평가대상

프로그램 정상 실행, 출력된 파일 내용(REPORT.TXT),
파일의 실시간 처리(장애 알림 프로그램이 출력하는 REPORT.TXT를 통해 실시간 처리 여부 검증)

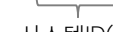
문제

4. 위 3번 문항까지 구현된 내용을 기준으로, 각각의 '모니터링 데이터(실시간)'별로 '시스템 장애'가 판별되는 경우 '장애 원인 프로세스'를 검출하여 '장애 알림 서버'로 송신하고, 수신된 응답에 따라 '장애 원인 프로세스'를 재전송하거나 '장애 현황 리포트'를 파일로 출력하시오.(20점)
- 입력 파일 개수 변경 : '※ 모니터링 데이터(실시간) 파일 복수 개로 변경' 참조
 - 장애 원인 프로세스 검출 기능 추가 :
 - '※ 장애 원인 프로세스 검출 규칙', '※ 장애 원인 프로세스 형식 정보' 참조
 - 장애 알림 프로그램 호출을 장애 알림 서버 통신으로 변경 :
 - '※ 장애 알림 서버 규격', '※ 구현 프로그램 통신 규칙' 참조
 - 장애 현황 리포트 기능 추가 : 장애 알림 결과가 "U"(Unknown)로 수신되면 장애 현황을 파일로 출력
 - '※ 장애 현황 리포트 파일 형식' 참조

상세설명

※ 모니터링 데이터(실시간) 파일 복수개로 변경

- 제공 프로그램(CREATE_MONITORING.EXE)은 5개의 '모니터링 데이터(실시간)' 파일을 생성함
- 제공 프로그램 실행 시 INPUT 폴더의 모든 파일들이 삭제된 후, 각 파일들은 데이터가 최초 추가되는 시점에
신규로 생성되고 데이터는 지속적으로 파일 끝에 추가됨
- 파일명 : MONITORING_{시스템ID(7)}.TXT (INPUT 폴더 내, 시스템ID는 임의의 영문자/숫자로 구성)
- 데이터 포맷 : '모니터링 데이터' 포맷과 동일(1번 문항과 동일)
- 입력데이터 파일 예시 : **MONITORING_LGWAS02.TXT**



시스템ID(7)

상세설명
(계속)

- ※ 장애 원인 프로세스 검출 규칙
- '시스템 장애' 발생 시 장애가 발생한 데이터를 포함한 **6건의 '추세 분석용 데이터'**를 대상으로 '장애 원인 프로세스' 검출
 - 시스템의 '이동 평균값' 증가/감소 추세와 '프로세스자원사용률' **증가/감소 추세가 동일한 프로세스** (동일한 프로세스가 없는 경우 **원인불명**으로 판단)
 - 위의 조건을 만족하는 프로세스 중,
시스템의 '이동 평균값'의 증가/감소 변화량 합에서 '프로세스자원사용률' 증가/감소 **변화량 합을 빼 결과의 절대값이 가장 적은 프로세스** (결과값이 동일한 프로세스는 없음)
 - '장애 원인 프로세스' 검출 예시 (0009 순번에서 '시스템 장애' 발생 예시)

6건의 '추세 분석용 데이터'

시스템 장애

순번	이동 평균값	프로세스 리소스 사용률					
		PROC01	PROC02	PROC03	PROC04	PROC05	
0004	072 증가 +2	024	002	005 증가 +2	000	004	
0005	074 동일 0	029	003	007 동일 0	001	000	
0006	074 증가 +3	029	003	007 증가 +8	001	014	
0007	077 증가 +17	033	015	015 증가 +3	002	012	
0008	094 감소 -2	038	017	018 감소 -2	005	009	
0009	092	038	017	016	004	015	

추세비교	증동증증감	증동증증동	증동증증동	증동증증감	증동증증감	감증감감증
변화량 합	20			11	4	
변화량 합의 차의 절대값				9(20-11)	16(20-4)	

추세가 동일하고 변화량 합의 차의 절대값이 가장 적음
→ **PROC03**이 장애 원인 프로세스

- ※ 장애 원인 프로세스 형식 정보
- 데이터 포맷 : '추세 분석용 데이터' 포맷 + **"#" + 장애 원인 프로세스이름(6, 원인불명 시 "NOTFND")**
 - 데이터 예시

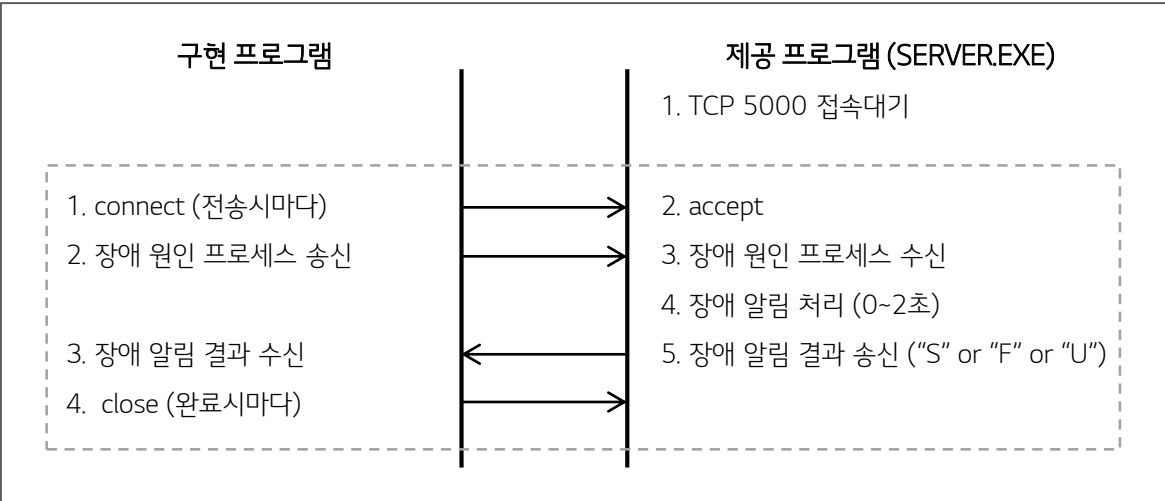
```
0009#SYS:090#PROC01:038#...#PROC05:015#Y#092#PROC03
0009#SYS:090#PROC01:038#...#PROC05:015#Y#092#NOTFND
```

상세설명
(계속)

- ※ 장애 알림 서버 규격 (제공 프로그램)
- 제공 프로그램(SERVER.EXE)를 실행하면 5000번 포트로 접속 대기(LISTEN)
 - 구현 프로그램으로부터 '장애 원인 프로세스' 데이터를 수신하여 장애 알림 처리(0~2초 소요) 후 결과 송신
 - 수신한 '장애 원인 프로세스' 데이터를 가공 없이 파일로 출력 (REPORT.TXT)
 - 통신 데이터 형식 정보 (서버 기준)
 - 수신 데이터 형식 : 시스템ID(7) + "#" + '장애 원인 프로세스' 포맷 + "\n"
 - 송신 데이터 형식 : 장애 알림 결과(1) + "\n"
- (모든 송수신 데이터의 끝은 LF(0x0A, '\n')로 종료)
- 장애 알림 결과: "S"(성공), "F"(실패), "U"(Unknown)
 - 콘솔에서 SERVER.EXE를 실행시킨 이후, 구현 프로그램을 실행해야 함
 - 제공 프로그램 실행 예시 :

C:\>SERVER.EXE<엔터키> ← SUPPORT 폴더에서 제공 프로그램 실행
...

※ 장애 알림 통신 예시



- ※ 구현 프로그램 통신 규칙
- '모니터링 데이터(실시간)' 파일에 대한 모니터링은 장애 알림 통신과 **별개로 실시간 처리 지속**
 - '시스템 장애' 판별 시 '장애 원인 프로세스'를 검출하여 **즉시** 장애 알림 통신 수행
 - '**시스템 장애**' 발생 순서대로 '장애 원인 프로세스' 데이터를 전송
 - 송/수신 완료 후 반드시 **Socket close** 필요
 - 수신한 장애 알림 결과가 "**F**"(실패)면 '장애 원인 프로세스' 데이터를 **재전송**함 (재전송은 1회만 수행, 재전송 시에도 "F"(실패)가 수신되면 재전송 하지 않음)
 - 수신한 장애 알림 결과가 "**U**"(Unknown)이면 새로운 출력 파일을 생성하여 **시스템ID 별로 마지막 판별된 '시스템 장애'**의 '장애 원인 프로세스' 데이터를 기록 후 파일 close ('※ 장애 현황 리포트 파일 형식 정보' 참조)

상세설명
(계속)

- ※ 장애 현황 리포트 파일 형식 정보
- 파일명 : REPORT_{순번(3)}.TXT (OUTPUT 폴더 내)
 - 순번은 "001"부터 시작하여 1씩 증가
 - 예> REPORT_001.TXT, REPORT_002.TXT, REPORT_003.TXT, ...
 - 데이터 포맷 : 시스템ID(7) + "#" + '장애 원인 프로세스' 포맷 + "\n"
 - 시스템ID 오름차순으로 출력
 - 출력 파일 예시

시스템ID
오름차순
출력

```
CNS0004#0009#SYS:090#PROC01:038#...#PROC05:015#Y#092#PROC03
KBWAS02#0009#SYS:077#PROC01:026#...#PROC05:002#Y#075#PROC01
KORDB07#0009#SYS:071#PROC01:007#...#PROC05:035#Y#071#PROC04
```

평가대상

프로그램 정상 실행, 출력된 파일 내용(REPORT.TXT, REPORT_{순번}.TXT),
파일의 실시간 처리(제공 프로그램과 구현 프로그램이 출력하는 REPORT.TXT , REPORT_{순번}.TXT를
통해 실시간 처리 여부 검증)

문제

5. 위 4번 문항까지 구현된 내용을 기준으로, 콘솔 입력을 통해 임계값을 변경하고 새로운 '시스템 장애' 판별 규칙을 추가하시오.(10점)

- 임계값 변경 기능 추가 : '※ 임계값 변경 기능 추가' 참조

- '시스템 장애' 판별 패턴 추가 : '※ 시스템 장애 판별 규칙 추가' 참조

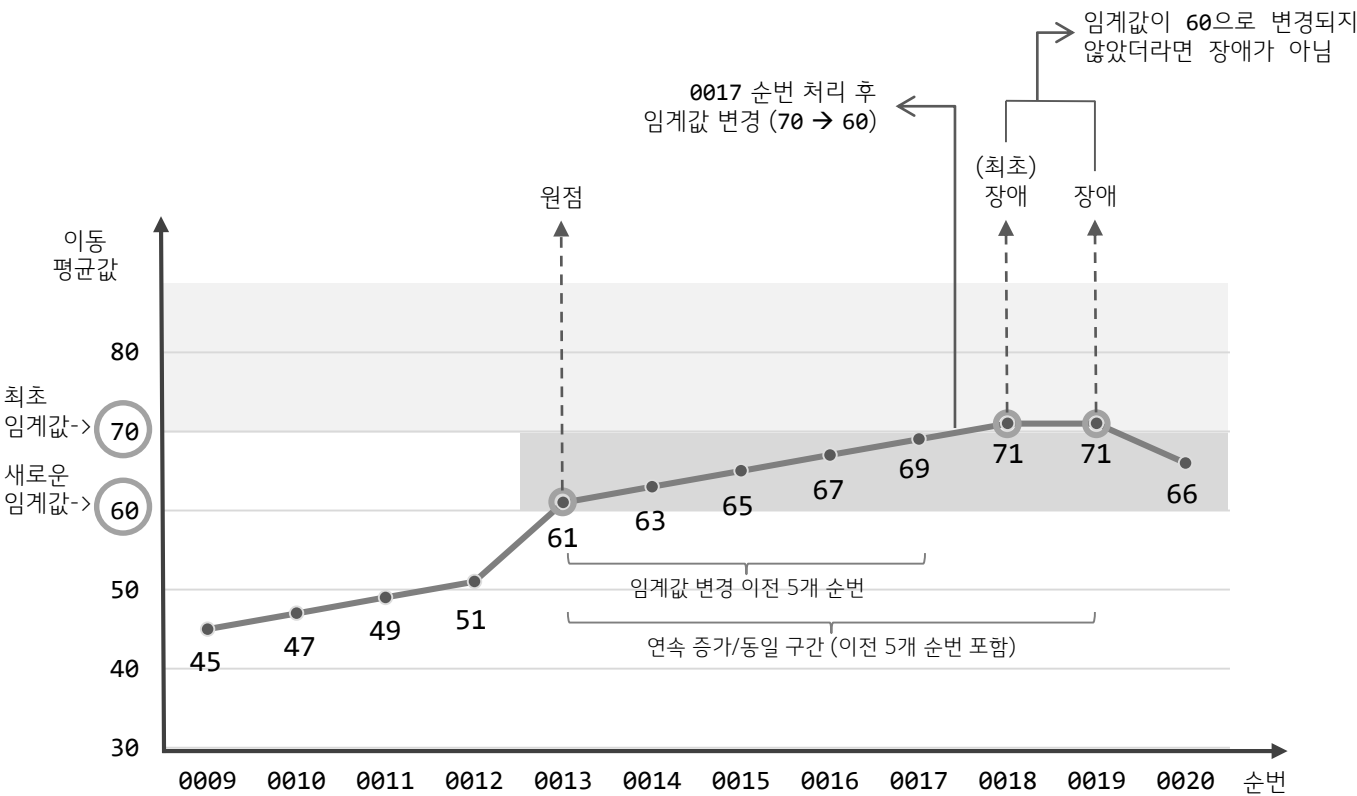
상세설명

- ※ 임계값 변경 기능 추가
- 구현 프로그램 실행 중 콘솔에 새로운 임계값을 입력하여 변경하는 기능 추가 (여러 번 변경 가능)

- 임계값이 변경되면 '시스템 장애' 상태는 초기화되고 새로운 임계값을 기준으로 이전 5개 순번부터 '시스템 장애' 판별
- 예> 0017 순번 처리 후 임계값 변경 → 0013 순번부터 새로운 임계값 초과에 따른 원점 판별
- 변경 예시 : 제공 프로그램(CREATE_MONITORING.EXE)의 지시에 따라 콘솔에 변경된 임계값 입력

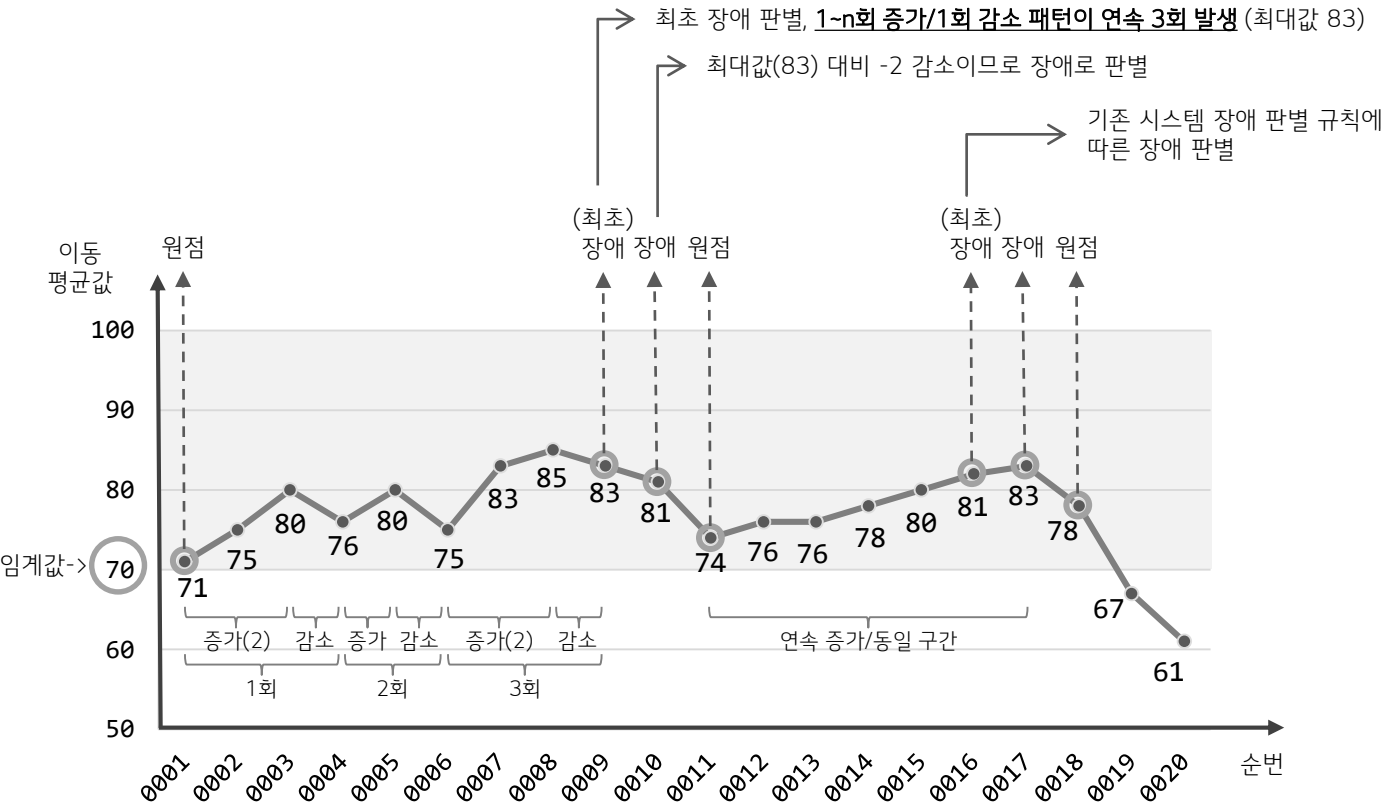
C:\>CREATE_MONITORING.EXE<엔터키>
...
임계값을 060으로 변경 후 엔터를 입력하세요 ← 1) 임계값 변경 지시 출력
PAUSE<엔터키> ← 3) 임계값 변경 후 엔터 입력
...
C:\>

C:\>SP_TEST<엔터키>
070<엔터키> ← 임계값 최초 입력
060<엔터키> ← 2) 임계값 변경 (임계값을 60으로 변경)
...



상세설명
(계속)

- ※ 시스템 장애 판별 규칙 추가
 - 기존 시스템 장애 조건에 새로운 최초 장애 판별 패턴 추가
 - > '임계값'을 초과한 이동 평균값의 **1~n회 증가/1회 감소 패턴이 연속 3회 발생**하는 경우 시스템 장애로 최초 판별
 - > 시스템 장애 최초 판별 이후에는, 최초 판별 이후 이동 평균값의 최대 값보다 -3이내 감소도 횟수에 포함 (기존 규칙과 동일)



평가대상

프로그램 정상 실행, 출력된 파일 내용(REPORT.TXT, REPORT_{순번}.TXT),
파일의 실시간 처리(제공 프로그램과 구현 프로그램이 출력하는 REPORT.TXT , REPORT_{순번}.TXT를
통해 실시간 처리 여부 검증)

폴더 정보

- ※ 프로그램 및 파일 위치 정보 (실행위치 기반 상대경로 사용 필수)
- 구현할 프로그램 위치 및 실행 위치 : 각 소문항 홈 (SUB1/SUB2/SUB3/SUB4/SUB5)
 - 모니터링 데이터 파일 폴더 위치 : 각 소문항 홈 아래 **INPUT** 폴더
(MONITORING.TXT, MONITORING_{시스템ID}.TXT)
 - 출력 파일 위치 : 각 소문항 홈 아래 **OUTPUT** 폴더 (REPORT.TXT, REPORT_{순번}.TXT)
 - 제공되는 모니터링 데이터 생성 프로그램 위치 및 파일명 :
SUPPORT폴더 내 CREATE_MONITORING.EXE
 - 제공되는 장애 알림 프로그램 위치 및 파일명 : **SUPPORT**폴더 내 ALERT.EXE
 - 제공되는 장애 알림 서버 프로그램 위치 및 파일명 : **SUPPORT**폴더 내 SERVER.EXE
 - 자가 검수용 참고 파일명 : OUTPUT폴더 내 CMP_*.TXT
- * 제공되는 파일들은 문항에 따라 다를 수 있음

실행 방식

- ※ 구현할 프로그램 형식
- 프로그램 형태 : 콘솔(Console) 프로그램
 - 프로그램 파일명 : SP_TEST
 - 실행 방식(문항1~2) : 콘솔 실행→결과처리→자동종료

C:\>SP_TEST<엔터키> ← 구현한 프로그램 실행 (Argument 없음)
070<엔터키> ← 콘솔 입력(임계값 70을 입력)
C:\>

- 실행방식(문항3~5) : 콘솔 실행→실시간 모니터링 (종료 없음)

C:\>SP_TEST<엔터키> ← 구현한 프로그램 실행 (Argument 없음)
070<엔터키> ← 콘솔 입력(임계값 70을 입력)
...

제공 및 제출

- ✓ 각 언어별 제공파일 압축 해제 후 자동 생성된 폴더 사용 필수
 - ✓ 제공되는 주요 내용
 - 입력 및 출력 샘플 파일
 - 모니터링 데이터 생성 프로그램 실행파일 (CREATE_MONITORING.EXE)
 - 장애 알림 프로그램 실행파일 (ALERT.EXE)
 - 장애 알림 서버 프로그램 실행파일 (SERVER.EXE)
 - 제출시 사용할 문항별 폴더 구조
 - ✓ 제출 파일 및 폴더 상세 내용 (각 언어별 실기 가이드 참고)
- <주의사항>
- 제출 파일 관련 내용 (폴더위치, 파일명, 프로그램명 등) 이 틀린 경우 및 상대경로를 사용하지 않은 경우에는 평가 시 불이익이 발생할 수 있으므로 반드시 요구되는 내용과 일치시켜 제출해야 함.

테스트 방법

※ 자가 검수를 위해 제공되는 샘플은 검수용 데이터와 다를 수 있음

검수를 위한 샘플 결과 파일은 각 문항 출력 폴더(OUTPUT)에 사전 제공됨

[문항1]

- SP_TEST를 실행(임계값 070 입력)한 후 출력 결과 파일(REPORT.TXT)을 샘플 결과 파일(CMP_REPORT.TXT)과 동일한지 비교

```
C:\>SP_TEST<엔터키>
070<엔터키>
C:\>
```

[문항2]

- SP_TEST를 실행(임계값 070 입력)한 후 출력 결과 파일(REPORT.TXT)을 샘플 결과 파일(CMP_REPORT.TXT)과 동일한지 비교

```
C:\>SP_TEST<엔터키>
070<엔터키>
C:\>
```

[문항3]

- SUPPORT 폴더에서 CREATE_MONITORING.EXE를 실행한 후, SP_TEST를 실행(임계값 070 입력)하고, 출력 결과 파일(REPORT.TXT)을 샘플 결과 파일(CMP_REPORT.TXT)과 동일한지 비교

```
C:\>CREATE_MONITORING.EXE<엔터키>
...
SP_TEST 실행 후 엔터를 입력하세요
PAUSE<엔터키>                                ← SP_TEST 실행 후 엔터 입력
...
C:\>
```

```
C:\>SP_TEST<엔터키>
070<엔터키>
...
```

테스트 방법
(계속)

- [문항4]
- SUPPORT 폴더에서 SERVER.EXE와 CREATE_MONITORING.EXE를 순차적으로 실행하고, SP_TEST를 실행(임계값 070 입력)
 - 출력 결과 파일(REPORT.TXT)을 샘플 결과 파일(CMP_REPORT.TXT)과 동일한지 비교
 - 출력 결과 파일(REPORT_{순번}.TXT)을 샘플 결과 파일(CMP_REPORT_{순번}.TXT)과 동일한지 비교

C:\>SERVER.EXE<엔터키>
...

C:\>CREATE_MONITORING.EXE<엔터키>
...
SP_TEST 실행 후 엔터를 입력하세요
PAUSE<엔터키> ← SP_TEST 실행 후 엔터 입력
...
C:\>

C:\>SP_TEST<엔터키>
070<엔터키>
...

- [문항5]
- SUPPORT 폴더에서 SERVER.EXE와 CREATE_MONITORING.EXE를 순차적으로 실행하고, SP_TEST를 실행(임계값 070 입력)
 - CREATE_MONITORING.EXE의 지시에 따라 SP_TEST 실행 콘솔에서 임계값 변경(임계값 060 입력)
 - 출력 결과 파일(REPORT.TXT)을 샘플 결과 파일(CMP_REPORT.TXT)과 동일한지 비교
 - 출력 결과 파일(REPORT_{순번}.TXT)을 샘플 결과 파일(CMP_REPORT_{순번}.TXT)과 동일한지 비교

C:\>SERVER.EXE<엔터키>
...

C:\>CREATE_MONITORING.EXE<엔터키>
...
SP_TEST 실행 후 엔터를 입력하세요
PAUSE<엔터키> ← SP_TEST 실행 후 엔터 입력
...
임계값을 060으로 변경 후 엔터를 입력하세요
PAUSE<엔터키> ← 임계값 변경 후 엔터 입력
...
C:\>

C:\>SP_TEST<엔터키>
070<엔터키> ← 임계값 최초 입력
060<엔터키> ← 임계값 변경
...

