

0번)



jun — -zsh — 80x24

```
Last login: Sat Nov  8 14:36:42 on ttys000
[(base) jun@hongseongjun-ui-MacBookAir ~ % mkdir shell-hw
[(base) jun@hongseongjun-ui-MacBookAir ~ % cd shell-hw
[(base) jun@hongseongjun-ui-MacBookAir shell-hw % touch ex3-0.sh
[(base) jun@hongseongjun-ui-MacBookAir shell-hw % nano ex3-0.sh
[(base) jun@hongseongjun-ui-MacBookAir shell-hw % echo $SHELL
/bin/zsh
[(base) jun@hongseongjun-ui-MacBookAir shell-hw % cd ~
[(base) jun@hongseongjun-ui-MacBookAir ~ % echo 'export MYENV="HELLO Shell"' >> ~/.zshrc
[(base) jun@hongseongjun-ui-MacBookAir ~ % source ~/.zshrc
[(base) jun@hongseongjun-ui-MacBookAir ~ % echo $MYENV
HELLO Shell
[(base) jun@hongseongjun-ui-MacBookAir ~ % unset MYENV
[(base) jun@hongseongjun-ui-MacBookAir ~ % echo $MYENV
zsh: command not found: ehco
[(base) jun@hongseongjun-ui-MacBookAir ~ % echo $MYENV

(base) jun@hongseongjun-ui-MacBookAir ~ %
```

사용한 명령어는 전부 캡쳐에 있다.

1번)



jun — -zsh — 80x24

```
Last login: Sun Nov  9 16:49:54 on ttys000
[(base) jun@hongseongjun-ui-MacBookAir ~ % nano ex3-1.sh
[(base) jun@hongseongjun-ui-MacBookAir ~ % chmod +x ex3-1.sh
[(base) jun@hongseongjun-ui-MacBookAir ~ % ./ex3-1.sh 7 3
나 늄 셈 : 2
덧 셈 : 10
뺄 셈 : 4
곱 셈 : 21
(base) jun@hongseongjun-ui-MacBookAir ~ %
```

nano에서 짠 코드는 아래와 같다.

```
#!/bin/bash
```

```
num1=$1
```

```
num2=$2
```

```
if [ -z "$num1" ] || [ -z "$num2" ]; then
```

```
echo "사용법: ./ex3-1.sh 숫자1 숫자2"
```

```
exit 1
```

```
fi
```

```
sum=$((num1 + num2))
```

```
diff=$((num1 - num2))
```

```
prod=$((num1 * num2))
```

```
if [ "$num2" -eq 0 ]; then
```

```
echo "0으로는 나눌 수 없다!"
```

```
else
```

```
div=$((num1 / num2))
```

```
echo "몫: $div"
```

```
fi
```

```
echo "덧셈: $sum"
```

```
echo "뺄셈: $diff"
```

```
echo "곱셈: $prod"
```

2번)

```
Last login: Sun Nov  9 16:57:33 on ttys000
[(base) jun@hongseongjun-ui-MacBookAir ~ % nano ex3-2.sh
[(base) jun@hongseongjun-ui-MacBookAir ~ % chmod +x ex3-2.sh
[(base) jun@hongseongjun-ui-MacBookAir ~ % ./ex3-2.sh
x 값을 입력하세요 : 5
x 값을 입력하세요 : 5
추가 x 값을 입력하려면 그냥 입력, 끝내려면 엔터 :
x=5 -> y=12.5
x=5 -> y=12.5
(base) jun@hongseongjun-ui-MacBookAir ~ %
```

nano에서 짠 코드는 아래와 같다.

```
#!/bin/bash
```

```
numbers=()
```

```
while [ ${#numbers[@]} -lt 2 ]; do
    read -p "x 값을 입력하세요: " input
    numbers+=("$input")
done
```

```
while true; do
    read -p "추가 x 값을 입력하려면 그냥 입력, 끝내려면 엔터: " input
    if [ -z "$input" ]; then
```

```

break
fi
numbers+="$input"
done

for x in "${numbers[@]}"; do
if [[ ! "$x" =~ ^-[0-9]+(\.[0-9]+)?$ ]]; then
echo "'$x'은 숫자가 아니어서 건너뜀"
continue
fi
y=$(echo "scale=6; 0.5 * ($x * $x)" | bc -l)
echo "x=$x -> y=$y"
done

```

3번)



```

Last login: Sun Nov  9 17:03:44 on ttys000
(base) jun@hongseongjun-ui-MacBookAir ~ % nano ex3-3.sh
(base) jun@hongseongjun-ui-MacBookAir ~ % chmod +x ex3-3.sh
(base) jun@hongseongjun-ui-MacBookAir ~ % ./ex3-3.sh
점수를 입력하세요 (0~100): 98
점수를 입력하세요 (0~100): 80
다른 점수를 입력하려면 입력, 끝내려면 엔터 :
점수 : 98 → 등급 : A
점수 : 80 → 등급 : B
평균 등급 : A
(base) jun@hongseongjun-ui-MacBookAir ~ %

```

nano에서 짠 코드는 아래와 같다.

```
#!/bin/bash

scores=()

while [ ${#scores[@]} -lt 2 ]; do

    read -p "점수를 입력하세요 (0~100): " s

    if [[ "$s" =~ ^[0-9]+\$ ]] && [ "$s" -ge 0 ] && [ "$s" -le 100 ]; then

        scores+=("$s")

    else

        echo "잘못된 입력, 0~100 사이 정수만!"

    fi

done

while true; do

    read -p "다른 점수를 입력하려면 입력, 끝내려면 엔터: " s

    [ -z "$s" ] && break

    if [[ "$s" =~ ^[0-9]+\$ ]] && [ "$s" -ge 0 ] && [ "$s" -le 100 ]; then

        scores+=("$s")

    else

        echo "잘못된 입력, 0~100 사이 정수만!"

    fi

done

grades=()

for s in "${scores[@]}"; do

    if [ "$s" -ge 90 ]; then

        g="A"

    else


```

```
g="B"  
fi  
grades+=("$g")  
echo "점수: $s → 등급: $g"  
done  
A_count=0  
for g in "${grades[@]}"; do  
[ "$g" = "A" ] && ((A_count++))  
done  
len=${#grades[@]}  
if [ "$A_count" -ge $((len/2+len%2)) ]; then  
avg_grade="A"  
else  
avg_grade="B"  
fi  
echo "평균 등급: $avg_grade"
```

4번)

```

=====
번호를 입력하세요 : 3
평균 점수 : 95.00

=====
1) 과목 성적 추가
2) 입력된 모든 점수 보기
3) 평균 점수 확인
4) 평균 등급(GPA) 변환
5) 종료

=====
번호를 입력하세요 : 4
평균 등급(GPA): A

=====
1) 과목 성적 추가
2) 입력된 모든 점수 보기
3) 평균 점수 확인
4) 평균 등급(GPA) 변환
5) 종료

=====
번호를 입력하세요 : 5
프로그램 종료
(base) jun@hongseongjun-ui-MacBookAir ~ %

```

nano에서 짠 코드는 아래와 같다.

```

#!/bin/bash

scores=()

while true; do

echo "====="

echo "1) 과목 성적 추가"
echo "2) 입력된 모든 점수 보기"
echo "3) 평균 점수 확인"
echo "4) 평균 등급(GPA) 변환"
echo "5) 종료"

echo "====="

read -p '번호를 입력하세요: ' choice

if [ "$choice" -eq 1 ]; then

read -p "점수(0~100)를 입력: " s

```

```

if [[ "$s" =~ ^[0-9]+\$ ]] && [ "$s" -ge 0 ] && [ "$s" -le 100 ]; then
    scores+=("$s")
    echo "성적이 추가됐습니다."
else
    echo "잘못된 점수!"
fi

elif [ "$choice" -eq 2 ]; then
    echo "입력된 점수: ${scores[@]}"
elif [ "$choice" -eq 3 ]; then
    if [ ${#scores[@]} -eq 0 ]; then
        echo "점수가 없음!"
    continue
fi

sum=0
for s in "${scores[@]}"; do
    sum=$((sum+s))
done
avg=$(echo "scale=2; $sum / ${#scores[@]}" | bc)
echo "평균 점수: $avg"

elif [ "$choice" -eq 4 ]; then
    if [ ${#scores[@]} -eq 0 ]; then
        echo "점수가 없음!"
    continue
fi

sum=0

```

```
for s in "${scores[@]}"; do
    sum=$((sum+s))
done
avg=$(echo "scale=2; $sum / ${#scores[@]}" | bc)
if (( $(echo "$avg >= 90" | bc) )); then
    gpa="A"
else
    gpa="B"
fi
echo "평균 등급(GPA): $gpa"
elif [ "$choice" -eq 5 ]; then
    echo "프로그램 종료"
    break
else
    echo "잘못된 입력! 1-5 중에서 골라야함!"
fi
echo
```

done
5번)



jun — -zsh — 80x24

```

/Applications/Br
jun      92186  0.0  0.2 435434720  35424 ?? S 5:27PM 0:00.80
/System/Library/
jun      92160  0.0  0.1 435360464  22400 ?? S 5:27PM 0:00.12
/System/Library/
jun      91325  0.0  0.1 435355920  9440 ?? S 5:16PM 0:00.14
/System/Library/
jun      90727  0.0  0.0 435305120  2704 ?? Ss 5:07PM 0:00.02
/Library/Apple/S
root    90723  0.0  0.0 435305360  3072 ?? Ss 5:07PM 0:00.02
/System/Library/
root    90722  0.0  0.0 435305168  2752 ?? Ss 5:07PM 0:00.05
/usr/libexec/usb
jun      90721  0.0  0.3 435528464  45552 ?? S 5:07PM 0:00.44
/System/Library/
jun      90714  0.0  0.1 435360320  17312 ?? S 5:07PM 0:00.17
/System/Library/
jun      90395  0.0  0.0 435366784  2336 ?? Ss 5:03PM 0:00.01
/System/Library/
jun      90389  0.0  0.2 435401968  34976 ?? Ss 5:03PM 0:00.41
/System/Library/
jun      90272  0.0  0.1 435308032  19088 ?? Ss 5:01PM 0:00.19
/System/Library/
(base) jun@hongseongjun-ui-MacBookAir ~ %

```

위 출력은 리눅스 시스템에서 실행중인 모든 프로세스 정보가 뜨게끔 하는 ps aux 명령어를 인자로 전달한 상태.

nano에서 짠 코드는 아래와 같다.

```

#!/bin/bash

run_cmd() {
    cmd=$1
    opts=$2
    eval "$cmd $opts"
}

run_cmd "$1" "$2"

```

6번)



shell-hw — zsh — 80x24

```
Last login: Sun Nov  9 17:28:56 on ttys000
[(base) jun@hongseongjun-ui-MacBookAir ~ % cd shell-hw
[(base) jun@hongseongjun-ui-MacBookAir shell-hw % touch hello.c
[(base) jun@hongseongjun-ui-MacBookAir shell-hw % nano hello.c
[(base) jun@hongseongjun-ui-MacBookAir shell-hw % gcc hello.c -o hello
[(base) jun@hongseongjun-ui-MacBookAir shell-hw % ./hello 오픈 소스 기초 설계 C 과제
```

시작

입력 인자 1: 오픈 소스 기초 설계

입력 인자 2: C

입력 인자 3: 과제

끝

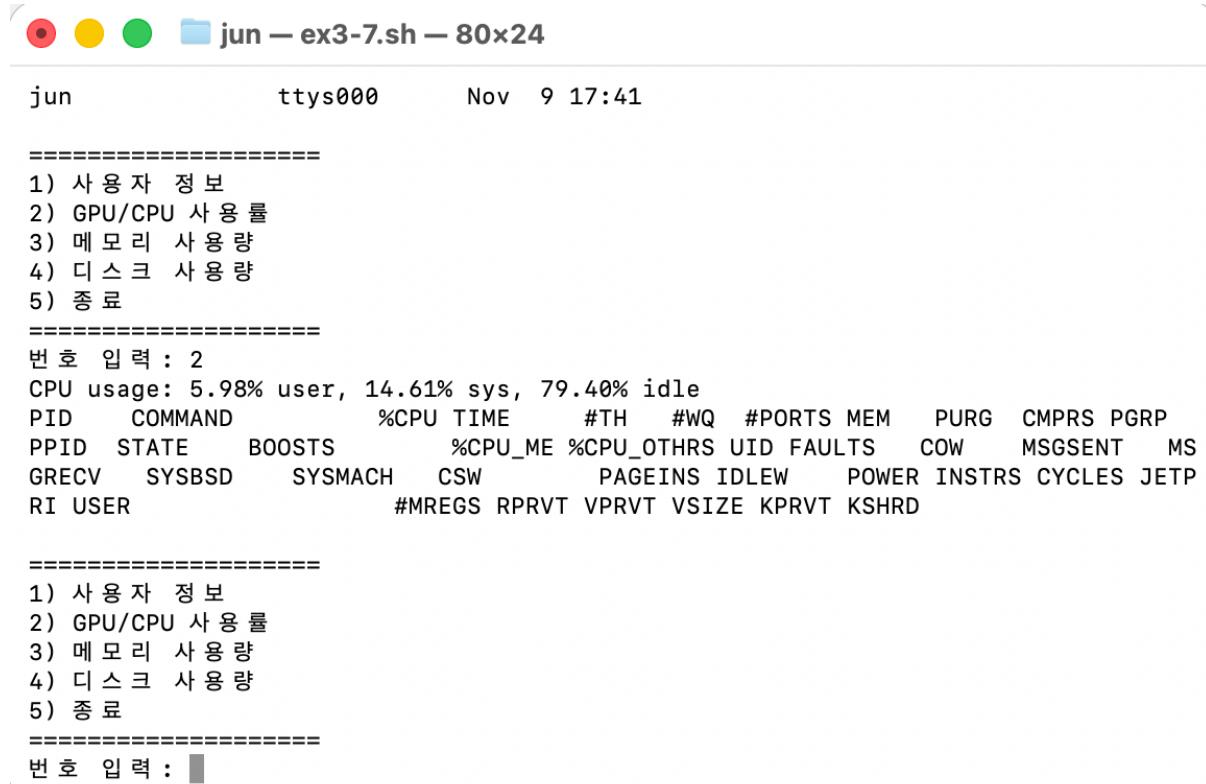
```
(base) jun@hongseongjun-ui-MacBookAir shell-hw %
```

nano에서 짠 코드는 아래와 같다.

```
#include <stdio.h>

int main(int argc, char *argv[]) {
    printf("시작\n");
    if (argc < 2) {
        printf("인자가 없습니다\n");
    } else {
        for (int i = 1; i < argc; i++) {
            printf("입력 인자 %d: %s\n", i, argv[i]);
        }
        printf("끝\n");
    }
    return 0;
}
```

7번)



```
jun      ttys000      Nov 9 17:41
=====
1) 사용자 정보
2) GPU/CPU 사용률
3) 메모리 사용량
4) 디스크 사용량
5) 종료
=====
번호 입력 : 2
CPU usage: 5.98% user, 14.61% sys, 79.40% idle
PID      COMMAND      %CPU TIME      #TH  #WQ  #PORTS MEM      PURG  CMPRS PGRP
PPID     STATE      BOOSTS      %CPU_ME %CPU_OTHRS UID FAULTS      COW  MSGSENT MS
GRECV   SYSBSD      SYSMACH    CSW      PAGEINS IDLEW      POWER INSTRS CYCLES JETP
RI USER          #MREGS RPRVT VPRVT VSIZE KPRVT KSHRD
=====
1) 사용자 정보
2) GPU/CPU 사용률
3) 메모리 사용량
4) 디스크 사용량
5) 종료
=====
번호 입력 :
```

nano에서 짠 코드는 아래와 같다.

```
#!/bin/bash

while true; do
    echo "====="
    echo "1) 사용자 정보"
    echo "2) GPU/CPU 사용률"
    echo "3) 메모리 사용량"
    echo "4) 디스크 사용량"
    echo "5) 종료"
    echo "====="
    read -p "번호 입력: " choice
```

```
if [ "$choice" == "1" ]; then
```

```
who

elif [ "$choice" == "2" ]; then
    if command -v nvidia-smi &> /dev/null; then
        nvidia-smi
    else
        top -bn1 | grep "%Cpu"
    fi

elif [ "$choice" == "3" ]; then
    free -h

elif [ "$choice" == "4" ]; then
    df -h

elif [ "$choice" == "5" ]; then
    echo "프로그램 종료"
    break

else
    echo "잘못된 선택입니다. 1~5 중 입력하세요."
    fi

echo
sleep 1

done

8번)
```

● ● ● train — zsh — 80x24

```
e3.txt file4.txt file5.txt
adding: file1.txt (stored 0%)
adding: file2.txt (stored 0%)
adding: file3.txt (stored 0%)
adding: file4.txt (stored 0%)
adding: file5.txt (stored 0%)
[(base) jun@hongseongjun-ui-MacBookAir DB % cd ~]
[(base) jun@hongseongjun-ui-MacBookAir ~ % cd .]
[(base) jun@hongseongjun-ui-MacBookAir ~ % mkdir -p train]
[(base) jun@hongseongjun-ui-MacBookAir ~ % cd train]
[(base) jun@hongseongjun-ui-MacBookAir train % ln -s ../DB/file1.txt file1.txt]
[(base) jun@hongseongjun-ui-MacBookAir train % ln -s ../DB/file2.txt file2.txt]
[(base) jun@hongseongjun-ui-MacBookAir train % ln -s ../DB/file3.txt file3.txt]

[(base) jun@hongseongjun-ui-MacBookAir train % ln -s ../DB/file4.txt file4.txt]
[(base) jun@hongseongjun-ui-MacBookAir train % ln -s ../DB/file5.txt file5.txt]
[(base) jun@hongseongjun-ui-MacBookAir train % ls -l]
total 0
lrwxr-xr-x 1 jun staff 15 Nov 9 17:50 file1.txt -> ../DB/file1.txt
lrwxr-xr-x 1 jun staff 15 Nov 9 17:51 file2.txt -> ../DB/file2.txt
lrwxr-xr-x 1 jun staff 15 Nov 9 17:51 file3.txt -> ../DB/file3.txt
lrwxr-xr-x 1 jun staff 15 Nov 9 17:51 file4.txt -> ../DB/file4.txt
lrwxr-xr-x 1 jun staff 15 Nov 9 17:51 file5.txt -> ../DB/file5.txt
(base) jun@hongseongjun-ui-MacBookAir train %
```

사용한 명령어는 아래와 같다.

mkdir -p DB

cd DB

touch file1.txt file2.txt file3.txt file4.txt file5.txt

zip files.zip file1.txt file2.txt file3.txt file4.txt file5.txt

cd .

mkdir -p train

cd train

ln -s ../DB/file1.txt file1.txt

ln -s ../DB/file2.txt file2.txt

ln -s ../DB/file3.txt file3.txt

ln -s ../DB/file4.txt file4.txt

ln -s ../DB/file5.txt file5.txt

ls -l

9번)

● ● ● jun — ex3-9.sh — 80x24

```
Last login: Sun Nov  9 17:47:03 on ttys000
[(base) jun@hongseongjun-ui-MacBookAir ~ % nano ex3-9.sh
[(base) jun@hongseongjun-ui-MacBookAir ~ % chmod +x ex3-9.sh
[(base) jun@hongseongjun-ui-MacBookAir ~ % ./ex3-9.sh
=====
1) 팀원 정보 추가
2) 팀원과 한 일 기록
3) 팀원 검색
4) 수행 내용 검색
5) 종료
=====
번호 입력 : 1
팀원 이름 : 홍길동
생일 (예 : 2000-01-01): 2000-01-01
전화 번호 : 010-1234-5678
추가 완료

=====
1) 팀원 정보 추가
2) 팀원과 한 일 기록
3) 팀원 검색
4) 수행 내용 검색
5) 종료
=====
```

nano에서 짠 코드는 아래와 같다.

```
#!/bin/bash

DB=DB.txt

while true; do

echo "====="

echo "1) 팀원 정보 추가"

echo "2) 팀원과 한 일 기록"

echo "3) 팀원 검색"

echo "4) 수행 내용 검색"

echo "5) 종료"

echo "====="

read -p "번호 입력: " menu
```

```
if [ "$menu" = "1" ]; then
    while true; do
        read -p "팀원 이름: " name
        [ -z "$name" ] // break
    done
    while true; do
        read -p "생일 (예: 2000-01-01): " birth
        [ -z "$birth" ] // break
    done
    while true; do
        read -p "전화번호: " phone
        [ -z "$phone" ] // break
    done
    echo "[INFO] $name,$birth,$phone" >> $DB
    echo "추가 완료"

elif [ "$menu" = "2" ]; then
    read -p "팀원 이름: " name
    read -p "날짜 (예: 2025-11-09): " date
    read -p "수행 내용: " job
    echo "[TASK] $name,$date,$job" >> $DB
    echo "기록 완료"

elif [ "$menu" = "3" ]; then
    read -p "검색할 팀원 이름: " name
```

```
grep "^[INFO] $name" $DB

elif [ "$menu" = "4" ]; then
    read -p "검색할 날짜(예: 2025-11-09) 또는 내용: " search
    grep "^[TASK].*$search" $DB

elif [ "$menu" = "5" ]; then
    echo "종료"
    break
else
    echo "잘못된 입력! 1~5 중에서 골라주세요."
fi

echo
sleep 1

done
```