

## **Навчальна практика з операційних систем та системного програмування**

Метою навчальної практики є формування у студентів здатності володіти знаннями щодо принципів роботи операційних систем, мати навички керування ресурсами обчислювальних систем, взаємодії з прикладним програмним забезпеченням, а також уміти обґрунтовано вибрати операційну систему для вирішення певних завдань і професійно її налаштувати, уміти керувати розподіленими ресурсами обчислювальної системи.

В результаті проходження практики студент повинен отримати знання та навички:

- з інсталяції і використання операційних систем GNU/Linux і Windows, в тому числі у віртуалізованих середовищах;
- з використання базового набору системних засобів операційної системи Linux для дослідження її поточного стану і керування обчислювальним процесом;
- з основ побудови операційних систем, їхньої архітектури, вимог до них, про історію їх розвитку і сучасні підходи до їх реалізації;
- про базовий склад компонентів операційної системи, основні функції ядра і системного програмного забезпечення;
- про методи і алгоритми керування локальними ресурсами комп'ютера: процесором, пам'яттю, пристроями введення-виведення, поділюваними ресурсами;
- про способи і засоби розв'язання проблем синхронізації і взаємних блокувань у багатозадачних і багатопотокових операційних системах;
- про принципи реалізації файлових систем, структуру сучасних файлових систем;
- про проблеми реалізації мережних функцій операційних систем і способи організації віддаленого виклику процедур і розподілених файлових систем;
- про підходи до реалізації зазначених вище механізмів у сучасних операційних системах GNU/Linux і MS Windows.

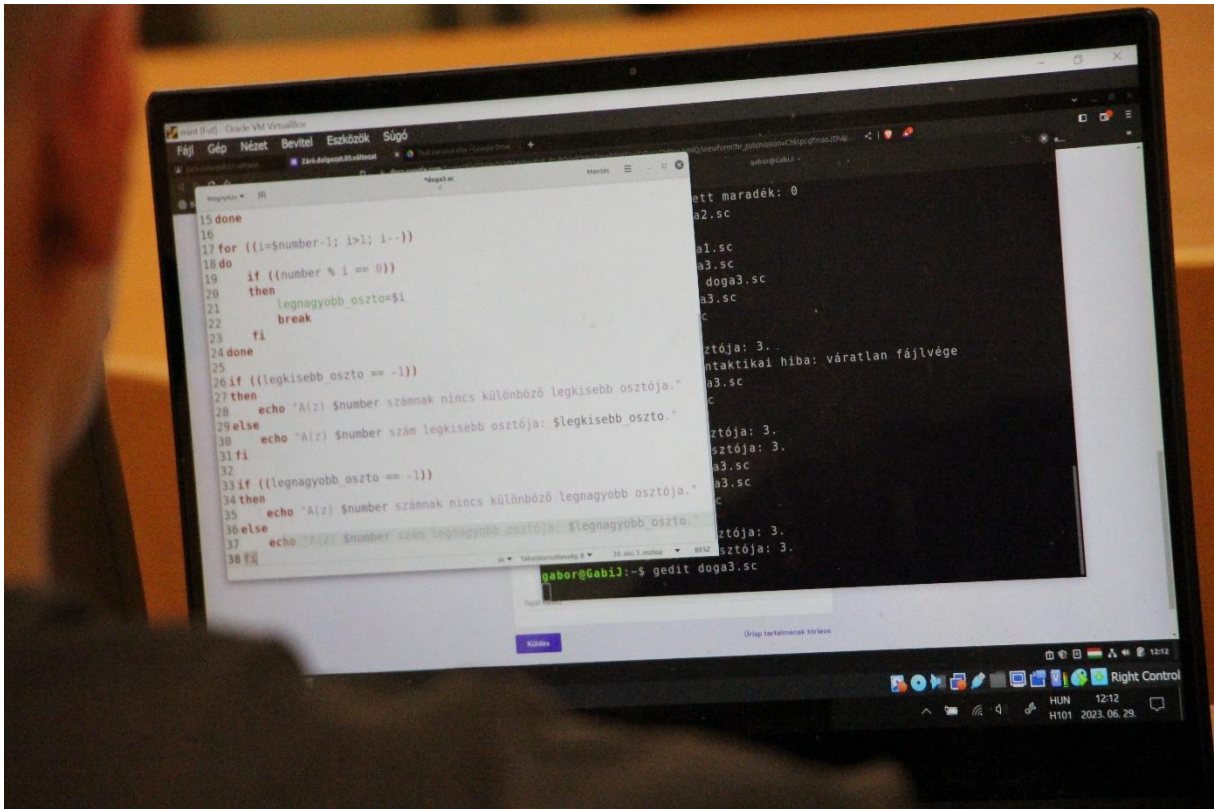
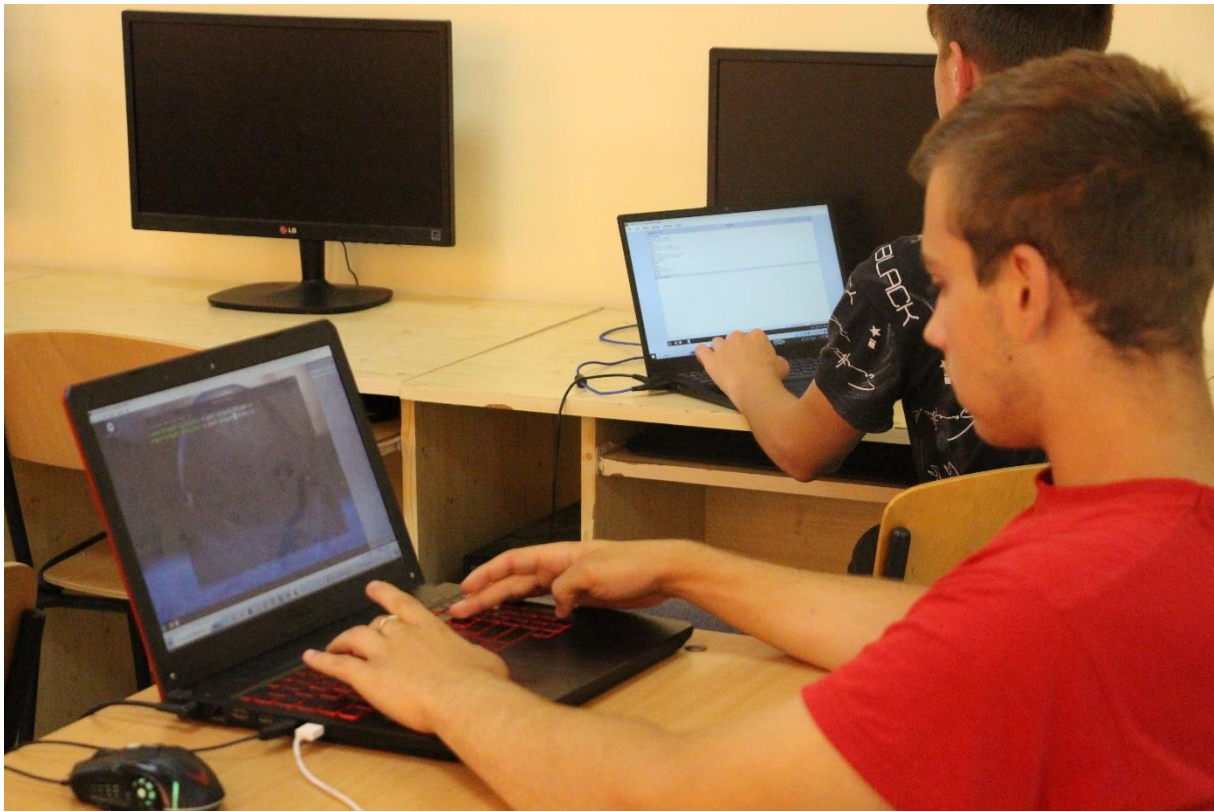
Під час практики студенти під керівництвом викладача знайомляться та/або опрацьовують різноманітні теми з операційних систем та системного програмування.

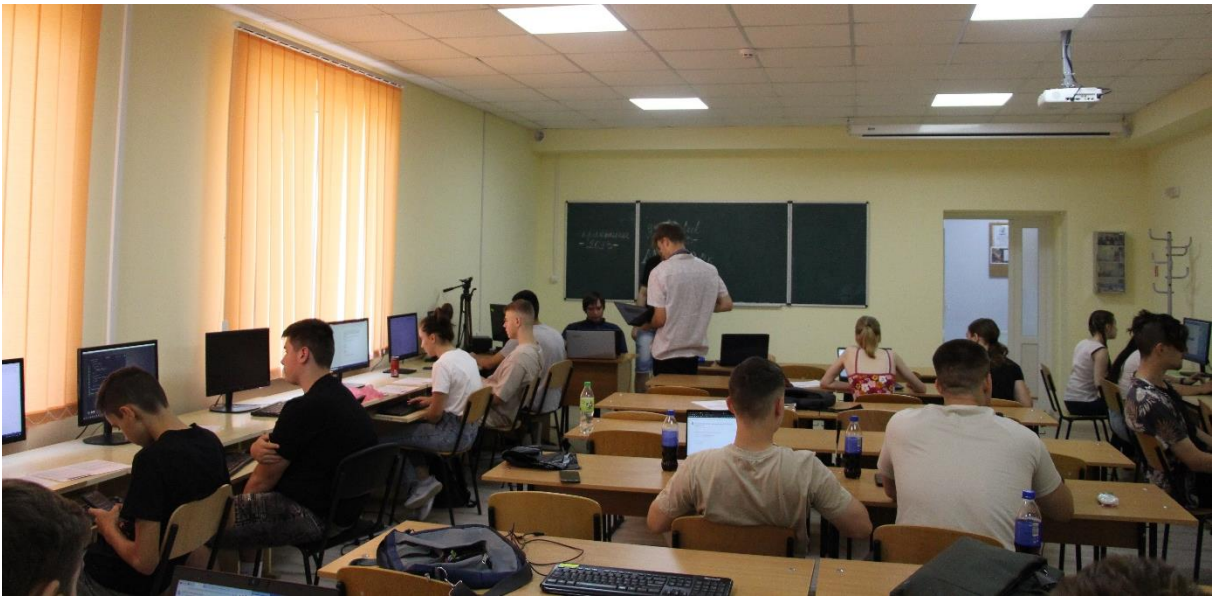
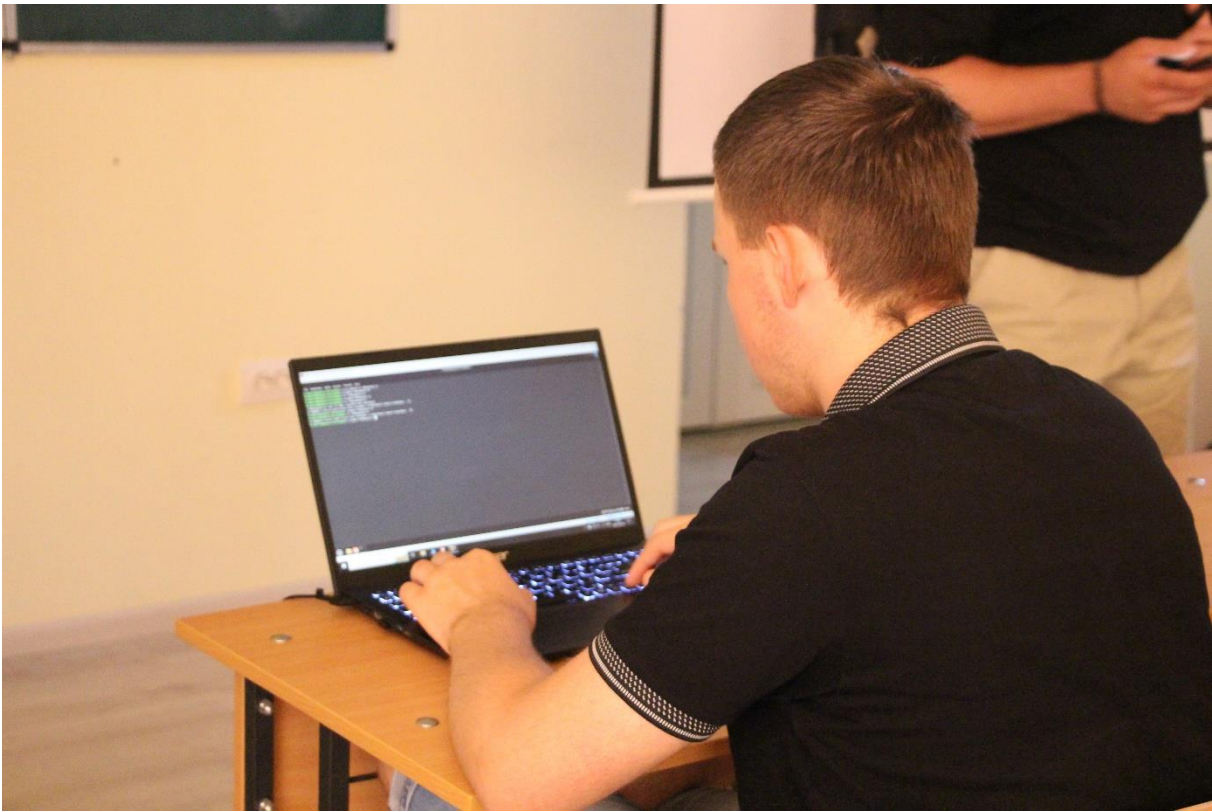
Орієнтовний зміст навчального дня практики:

- обговорення з керівником практики поточної теми;
- індивідуальне опрацювання спеціалізованих задач згідно плану практики;
- документальне оформлення ходу розв'язання завдань.

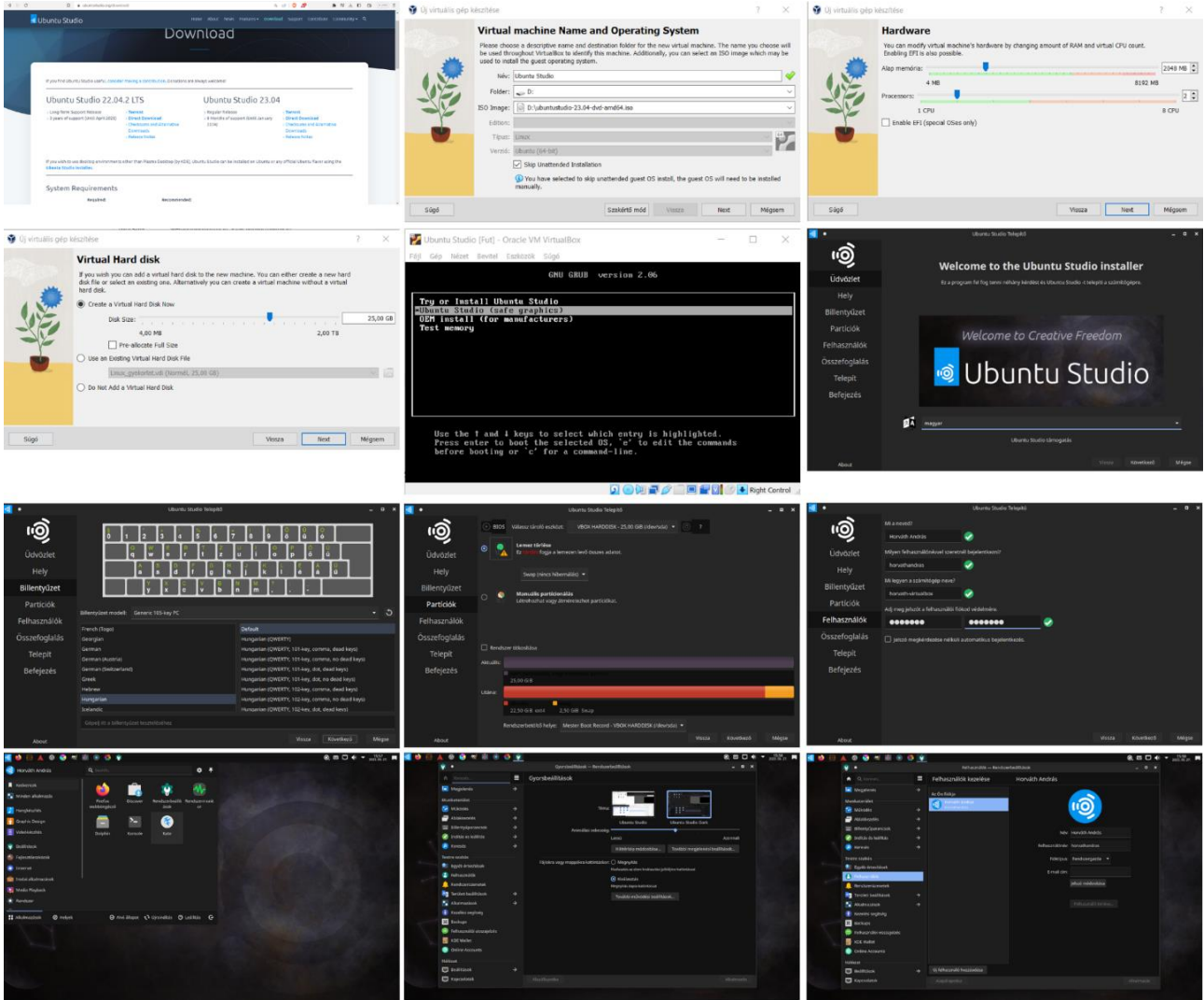
**Світлини з навчальної практики з операційних систем  
та системного програмування  
2022-2023**







# Из виконаних звітів робіт студентів:



```
#!/bin/bash
i=1
while [ $i -le $1 ]
do
do
j=1
szorzat=1
while [ $j -le $i ]
do
szorzat=$((szorzat*$j | bc -l))
j=$((j+1))
done
echo -n "$szorzat "
i=$((i+1))
done
echo

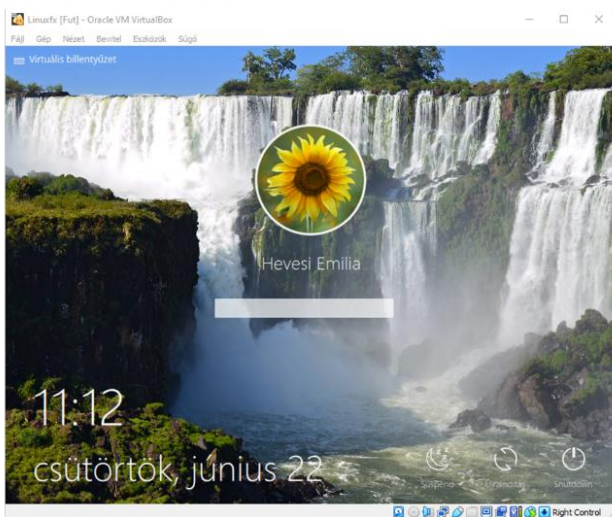
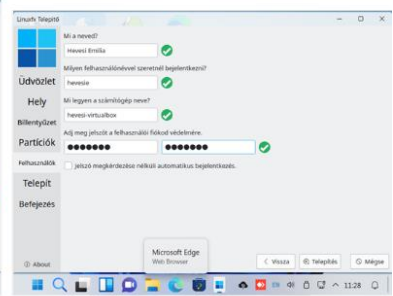
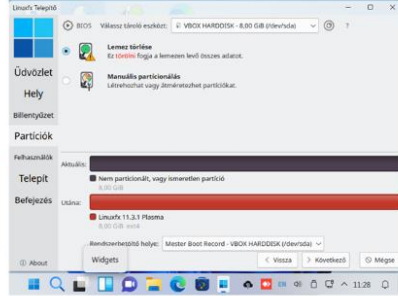
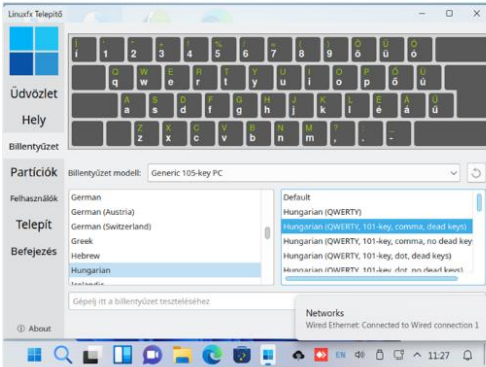
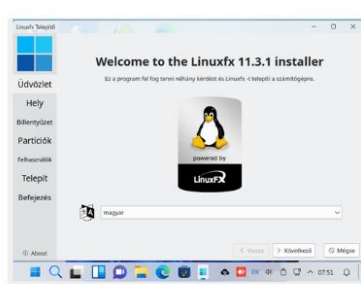
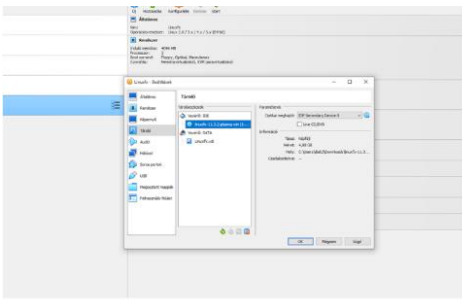
#!/bin/bash
read string1
read string2
if [ ${string1} -eq ${string2} ]
then
echo "A két stringnek ugyanaz az első karaktere"
elif [ ${string1} -lt ${string2} ]
then
echo "Az első string első karaktere van előbb az ASCII táblázatban"
else
echo "A második string első karaktere van előbb az ASCII táblázatban"
fi
else
echo "Nem ugyanaz a hossza a két stringnek"
fi

#!/bin/bash
S=$1
tomb=()
fuggveny ()
{
for (( i=0 ; i<$S ; i++ ))
do
let tomb[$i]=$RANDOM%41+0
done
p=0
for (( j=0 ; j<3 ; j++ ))
do
if [ ${tomb[$j]} -gt 0 ]
then
echo -n "${tomb[$j]}, pozíció: $j. "
let p=$((p+1))
fi
done
echo
}
fuggveny

#!/bin/bash
tomb=()
db=0
for (( i=0 ; i<100 ; i++ ))
do
let tomb[$i]=$RANDOM%41+0
if [ ${tomb[$i]} -eq 25 ]
then
let db=$((db+1))
fi
done
echo -n "A tömbben a 25-ös szám $db-szor/szer szerepel, a következő indexű elemekben: "
for (( i=0 ; i<100 ; i++ ))
do
if [ ${tomb[$i]} -eq 25 ]
then
echo -n " $i. "
fi
done
echo

#!/bin/bash
x=$1
if [ $x -lt -4 ]
then
a=$(echo "e($x)" | bc -l)
b=$(echo "l($x)" | bc -l)
echo $(echo "$a+$b" | bc)
elif [ $x -ge -4 ] && [ $x -lt 3 ]
then
a=$((($echo "-5*$x" | bc -l)+7))
b=$(echo "$x*$x" | bc -l)
echo $(echo "$b+$a" | bc -l)
elif [ $x -ge 3 ]
then
a=$(echo "s(($echo "$x*2" | bc -l))" | bc -l)
b=$(echo "$x+1" | bc -l)
echo $(echo "$a/$b" | bc -l)
fi

#!/bin/bash
i=1985
while [ $i -le 2000 ]
do
cal 12 $i
sleep 3
clear
i=$((i+1))
done
```



```
#!/bin/bash
n=0
echo "Adj meg egy pozitív egészt: "
while [ $n -lt 1 ]
do
    read n
    fib1=0
    fib2=1
    if [ $n -gt 0 ]
    then
        echo "Az első $n Fibonacci-szám: "
        m=`expr $n - 1`
        echo "$m"
        for i in `seq 1 1 $m`
        do
            let fib=fib1+fib2
            fib1=$fib2
            fib2=$fib
            echo "$fib"
        done
    else
        echo "A megadott érték nem megfelelő"
    fi
done

#!/bin/bash
db=0
declare -a Tomb[10]
echo -n "A tömb elemei:"
for i in `seq 0 1 9`
do
    k=`expr $RANDOM%151`
    let Tomb[$i]=k
    echo -n "${Tomb[$i]} "
    if [ ${Tomb[$i]} -eq 120 ]
    then
        let db=db+1
    fi
done
echo ""
if [ $db -gt 0 ]
then
    echo "A 120-as $db alkalommal szerepel a tömbben"
else
    echo "A tömb elemei közt nincs 120-as érték"
fi

#!/bin/bash
Tomb=(-1 3 2 -5 3 -8 7 -5)
declare -a S
echo -n "A tömb elemei: "
echo "${Tomb[@]}"
for i in `seq 7 -1 1`
do
    for n in `seq 0 1 6`
    do
        if [ ${Tomb[$n+1]} -gt ${Tomb[$n]} ]
        then
            a=${Tomb[$n]}
            let Tomb[$n]=Tomb[$n+1]
            let Tomb[$n+1]=a
        fi
    done
done
echo -n "A tömb elemei, rendezve: "
echo "${Tomb[@]}"

#!/bin/bash
if [ $# -eq 1 ] && [ $1 -ge 0 ]
then
    S=1
    for i in `seq 1 1 $1`
    do
        let S=S*i
    done
    echo "$S! = $S"
else
    echo "Egyetlen, nemnegatív nagyobb paramétert adj meg!"
fi

#!/bin/bash
echo -n "Adj meg egy mondatot: "
read mondat
echo "$mondat" | cat > mondat.txt
a=`grep -E 'm' mondat.txt | grep 'k' mondat.txt`
if [ -n $a ]
then
    echo "A mondat tartalmaz m-mel kezdődő és k-val végződő szót"
else
    echo "A mondat nem tartalmaz m-mel kezdődő és k-val végződő szót"
fi

#!/bin/bash
echo -n "Adj meg egy számot: "
read szam
k=`expr $szam % 2`
if [ $k -eq 0 ]
then
    echo "Paros"
else
    maradek=`expr $szam % 3`
    echo "Paratlan"
    echo "3-mal való osztási maradék: $maradek"
fi
```