

## **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.

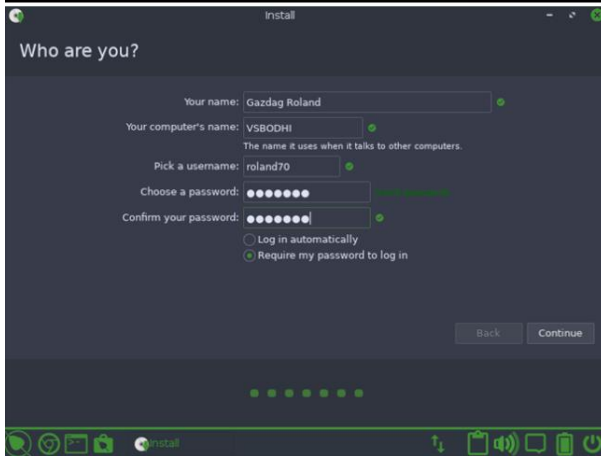
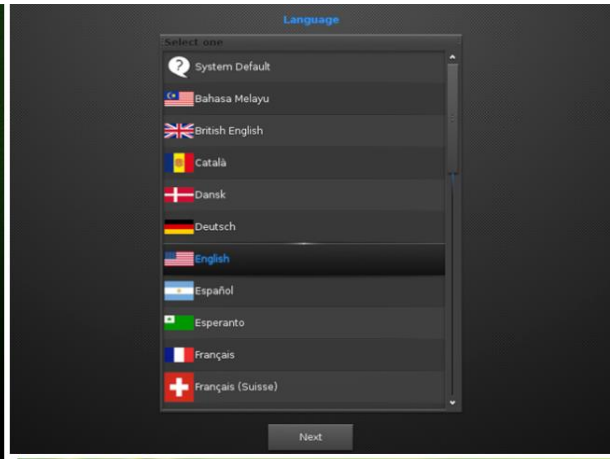
**Світлини з навчальної практики з операційних систем  
та системного програмування  
2020-2021**







## Iz виконаних звітів робіт студентів:



```

#!/bin/bash
a=0
read -p "Add meg a felhasználónevet : " login
if test "$login" = "user"
then
    getent passwd $1 > /dev/null
    if [ $? -eq 0 ];
    then
        echo "Van ilyen felhasználó"
        echo $(groups $login)
        for u in $(who | awk '{print $1}' | sort | uniq)
        do
            if [ "$u" = "$login" ]; then
                a=1
            fi
        done
        if [ "$a" = "1" ]; then
            echo "$login be van jelentkezve."
            exit 0
        else
            echo "$login nincs bejelentkezve"
            exit 1
        fi
        echo "A felhasználó a következőként van bejelentkezve: $(who -T )"
    else
        echo "Nincs ilyen felhasználó"
    fi
else
    echo "Nem ez a szükséges felhasználó, vagy nincs ilyen felhasználó a számítógépen."
fi

#!/bin/bash
if [ $1 -gt $2 ]
then
    szamol=$((($1*$2))
    echo "$szamol"
else
    szamol=$(expr "scale=4; (($1^2)*$2*$3)/($1+$2+$3)*s(4*(a(1)))" | bc -l)
    echo "$szamol"
fi

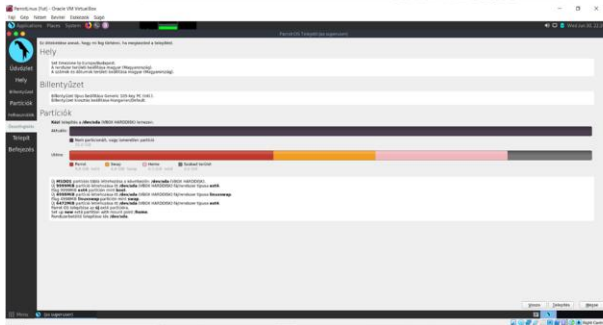
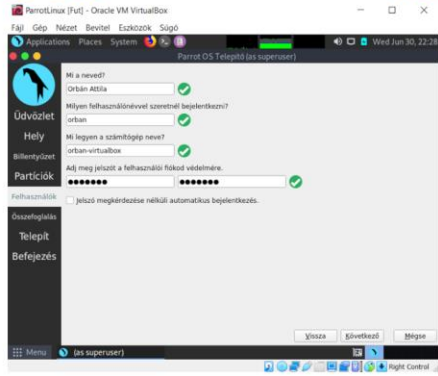
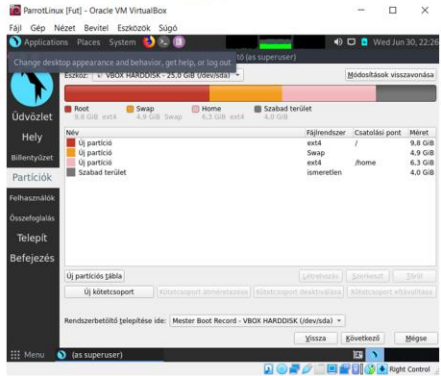
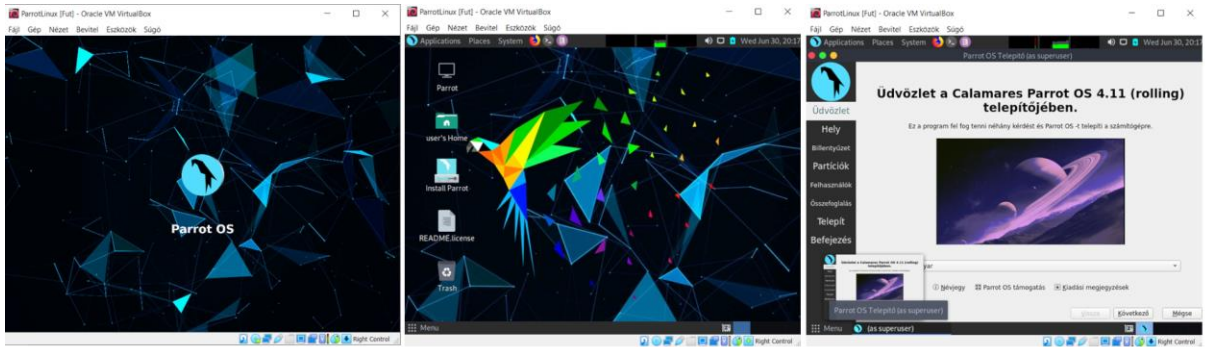
#!/bin/bash
kezd=1
for i in `seq 1 1 $1`
do
    kezd=$((kezd*$i))
done
echo "$1!=$kezd"

#!/bin/bash
egesz=$((($1%2))
if [ $egesz -eq 0 ] || [ $egesz -eq 5 ]
then
    if [ $1 -lt 50 ]
    then
        for j in `seq 1 1 $1`
        do
            x=1
            for i in `seq 1 1 $j`
            do
                x=$((x*$i))
            done
            echo "$x"
        done
    else
        echo "A szam nem kisebb mint 50"
    fi
else
    echo "Ez nem egesz szam"
fi

#!/bin/bash
for i in $1 $2 $3
do
    oszt=$((($i%3))
    if [ $oszt -eq 0 ]
    then
        echo "$i"
    fi
done

#!/bin/bash
n=$((($1-1))
tomb=()
for i in `seq 0 1 $n`
do
    echo "Add meg a tomb $i. elemét:"
    read tomb[$i]
done
max=${tomb[0]}
for i in `seq 0 1 $n`
do
    if [ ${tomb[$i]} -gt $max ]
    then
        max=${tomb[$i]}
    fi
done
echo "${tomb[*]}"
echo " A tomb legnagyobb eleme: $max"

```



```
#!/bin/bash
s=$1
if [ $2 -lt $s ]
then
s=$2
fi
if [ $3 -lt $s ]
then
s=$3
fi
echo "Legkisebb szám $1 $2 $3 között: $s"
fakt=1
while [ $s -gt 1 ]
do
fakt=$((fakt*$s))
s=$((s-1))
done
echo "A szám faktoriálja: $fakt"

#!/bin/bash
if [ $1 -lt 9 ]
then
if [ $1 -gt 2 ]
then
for i in `seq 1 100`
do
db=0
for n in `seq 1 1 $i`
do
osztó=$((i%n))
if [ $osztó -eq 0 ]
then
db=$((db+1))
fi
done
if [ $db -eq $1 ]
then
echo "$i"
fi
fi
fi

#!/bin/bash
osszeg=0
darab=0
for i in `seq $1 $2`
do
if [ (($i%3) -eq 0) ] && [ (($i%5) -eq 0) ]
then
echo "$i"
osszeg=$((osszeg+i))
darab=$((darab+1))
fi
done
echo "A számok összege: $osszeg"
echo "$darab szám osztható 3-al es 5-el"

#!/bin/bash
a=$1
b=$2
c=$3
if [ $a -gt $b ]
then
osztas=$((a/b))
echo "Az első szám másodikkal való osztásának maradéka: $osztas"
fi
muveletek= echo "expr `scale=4; (((a^2)*b*c)/($a+$b+$c))` | bc -l"
echo "$muveletek"

#!/bin/bash
a=$((1%2))
if [ $a -eq 0 ]
then
echo "A megadott szám paros"
else
b=$((1%3))
echo "A megadott paratlan szám 3-val valo osztásának maradéka: $b"
fi

#!/bin/bash
echo "A következő fájloknak van futtatási joga:"
for fajl in *
do
if [ -f $fajl ]
then
if [ -x $fajl ]
then
ls -l $fajl
fi
fi
done

#!/bin/bash
fakt=1
for i in `seq 1 1 $1`
do
fakt=$((fakt*i))
a=$((i-1))
done
echo "A $1!= $fakt"

#!/bin/bash
max=0
n=$1
tomb=()
for i in `seq 1 1 $n`
do
read tomb[$i]
if [ ${tomb[$i]} -gt $max ]
then
max=${tomb[$i]}
fi
done
echo "${tomb[*]}"
echo "A(z) $n elemu tomb legnagyobb eleme: $max"
```