

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

**Інститут ІКНІ  
Кафедра ПЗ**

**ЗВІТ**

до лабораторної роботи № 2

**На тему:** *“Ознайомлення з процесами в ОС Linux/Macos.”*

**З дисципліни:** *“Операційні системи”*

**Лектор:**

ст. викладач ПЗ

Грицай О.Д.

**Виконав:**

ст. гр. ПЗ-22

Ясногородський Н.В.

**Прийняв:**

ст. викладач ПЗ

Грицай О.Д.

« \_\_\_\_ » \_\_\_\_\_ 2022 р.

Σ= \_\_\_\_ .

**Тема роботи:** Ознайомлення та керування процесами в операційних системах для персонального комп'ютера. Linux та MacOS.

**Мета роботи:** Ознайомитися з процесами та потоками в операційних системах Linux, MacOS. Навчитися працювати із системними утилітами, що дають можливість отримувати інформацію про процеси, потоки, використовувану ними пам'ять, та іншу необхідну інформацію.

## Теоретичні відомості

**Операційна система** - це сукупність програм, які призначені для керування ресурсами комп'ютера й обчислювальними процесами, а також для організації взаємодії користувача з апаратурою.

**Перша функція ОС** - керування ресурсами комп'ютера та їх розподіл. Ресурси - це логічні й фізичні компоненти комп'ютера: оперативна пам'ять, місце на диску, периферійні пристрої, процесорний час тощо.

**Інша функція ОС** - керування обчислювальними процесами. Обчислювальним процесом (або завданням) називається послідовність дій, яка задається програмою. У принципі, функції керування процесами можна було б передати кожній прикладній програмі, але тоді програми були б набагато більшими та складнішими. Тому зручніше мати на комп'ютері одну керуючу програму - операційну систему, послугами якої користуватимуться всі інші програми.

**Для виконання третьої функції ОС** - забезпечення взаємодії користувача з апаратурою - служить інтерфейс користувача ОС. До складу інтерфейсу користувача входить також набір сервісних програм - утиліт.

**Утиліта** - це невелика програма, що виконує конкретну сервісну функцію. Утиліти звільняють користувача від виконання рутинних і часом досить складних операцій.

**Процес** — одне з найважливіших понять у архітектурі операційних систем та програмуванні. Процес — об'єкт операційної системи, контейнер системних ресурсів, призначених для підтримки виконання програми.

## Індивідуальне завдання

1. Встановити операційні системи Linux та MacOS
2. За допомогою консольних засобів ОС Linux отримати повну інформацію про процеси.
3. За допомогою утиліт top, htop, qps, System Monitor отримати повну інформацію про процеси в ОС Linux та MacOS.
4. Використовуючи консольні засоби ОС Linux та утиліти змінити пріоритет виконання процесу.
5. Використовуючи консольні засоби ОС Linux та сторонні утиліти змінити стан виконання процесу, завершити виконання заданого процесу.
6. Скомпілювати файл main.cpp представлений у лабораторній роботі № 1 (на MacOS і Linux можна командою: g++ main.cpp -pthread) і запустити виконуваний файл на різних кількості активних процесорів (ядер). Знайти для даної програми величини A, S, p при різних вхідних значеннях величини n. Порівняти результати для різних операційних систем.

7. Результати лабораторної роботи оформити у звіт, у висновку надати порівняння моніторингу процесів у різних системах різними утилітами, відповідно до індивідуального варіанту.

## Протокол роботи

Я мав доступ до нативно встановлених macOS та GNU/Linux операційних систем, лабораторна виконувалась без віртуальних машин

```
ps x
ps x
PID TT STAT TIME COMMAND
548 ?? S 0:40.50 /usr/sbin/distnoted agent
549 ?? S 0:19.38 /usr/sbin/cfprefsd agent
556 ?? S 0:12.53 /usr/libexec/UserEventAgent (Aqua)
560 ?? S 0:05.05 /usr/libexec/lsd
561 ?? S 0:19.16 /usr/libexec/knowledge-agent
562 ?? S 0:00.52 /System/Library/PrivateFrameworks/CloudServices.framework/Helpers/com.apple.sbd
563 ?? S 0:02.50 /System/Library/Frameworks/CoreTelephony.framework/Support/CommCenter -L
564 ?? S 0:00.02 /System/Library/CoreServices/APFSUserAgent
565 ?? S 0:08.34 /System/Library/CoreServices/LogoutAgent
566 ?? S 0:07.75 /usr/libexec/nsurlsessiond
567 ?? S 0:00.02 /System/Library/Frameworks/CoreServices.framework/Versions/A/Frameworks/CarbonCore.framework/Versions/A/XPCServices/csnameddatad.xpc/Contents/MacOS/csnameddata
568 ?? S 0:01.04 /usr/sbin/ucrmnoted
569 ?? S 0:08.66 /usr/libexec/routrined LAUNCHED_BY_LAUNCHD
570 ?? S 0:04.51 /usr/libexec/rapportd
571 ?? S 0:06.13 /usr/libexec/secd
572 ?? S 0:00.40 /usr/libexec/secinitd
573 ?? S 0:00.83 /usr/libexec/smd
574 ?? S 0:29.30 /System/Library/PrivateFrameworks/ScreenTimeCore.framework/Versions/A/ScreenTimeAgent
575 ?? S 0:02.26 /System/Library/PrivateFrameworks/DoNotDisturbServer.framework/Support/donotdisturbd
576 ?? S 0:19.93 /System/Library/PrivateFrameworks/CoreDuetContext.framework/Resources/ContextStoreAgent
577 ?? S 0:06.54 /System/Library/PrivateFrameworks/IDS.framework/IdentityServicesd.app/Contents/MacOS/IdentityServicesd
578 ?? S 15:48.29 /Applications/Brave Browser.app/Contents/MacOS/Brave Browser
579 ?? S 0:05.62 /System/Library/PrivateFrameworks/TCC.framework/Support/tccd
```

Рис. 1. Отримання інформації про процеси за допомогою консольної утиліти *ps*

За допомогою утиліт *top*, *htop*, *Activity Monitor*, *qps* отримаю повну інформацію про процеси в ОС Linux та MacOS.

```
Processes: 564 total, 4 running, 560 sleeping, 2860 threads
Load Avg: 1.27, 1.32, 1.53 CPU usage: 6.21% user, 5.19% sys, 88.58% idle SharedLibs: 582M resident, 113M data, 248M linkedit.
MemRegions: 466875 total, 106 resident, 354M private, 3716M shared. PhysMem: 31G used (1952M wired), 426M unused. VM: 2511 vszize, 3831M framework vszize, 0(0) swapins, 0(0) swaptouts.
Networks: packets: 38983265/496 in, 2313241/392M out. Disks: 2158173/786 read, 2059551/1326 written.
21:44:29
PID COMMAND %CPU TIME #TH #WO #PORT MEM PURG CMPRS PGRP PPID STATE BOOSTS %CPU_ME %CPU_OTHRS UID FAULTS COW MSGSENT MSGRECV SYSBSD
378 WindowServer 55.8 01:48:13 21/1 5 2161 511M+ 15M- 25M 370 1 running #0(1) 0.05869 0.09724 88 5914253+ 8871 66999675+ 4752360+ 96556733+
0 kernel_task 19.9 06:14:73 523/10 0 0 38M 0 0 0 0 0 running #0(0) 0.00000 0.00000 0 32075 0 174970591+ 60399654+ 0
90848 screencapt 11.9 00:00:27 5 4 65 7921K+ 752K 0 593 593 sleeping #0(225+) 0.45022 0.00000 501 4463+ 553+ 2342+ 729+ 1808+
89987 top 6.6 00:00:08 1/1 0 28 6833K 0 0 89987 86765 running #0(1) 0.00000 0.00000 0 5187+ 72 1071706+ 535847+ 18787+
878 Brave Browse 5.0 11:38:28 33 1 386 322M+ 160K 35M 578 578 sleeping #0(8) 0.00000 0.00000 501 6773553+ 16177+ 3667918+ 1654475+ 7126299+
389 karabiner_gr 3.0 00:22:27 19 1 162 12M 0 6448K 389 1 sleeping #0(1) 0.00000 0.00000 0 58663 79 530532+ 2424+ 11817162+
693 yabai 2.0 01:26:40 5/2 2/1 145 24M 0 3540K 693 1 running #0(1) 0.00000 0.00000 501 279779+ 20226 106484+ 19724+ 3659041+
90847 screencapt 1.5 00:00:09 3 1 153 8242K 0 0 90847 1 sleeping #0(214+) 0.00000 0.45022 501 1814 164 952+ 384+ 2862+
86764 aLacrity 1.4 00:07:47 12 5 230 285M+ 192K+ 0 86764 1 sleeping #0(63) 0.00745 0.00000 501 12522 250 77974+ 79452+ 179540+
307 karabiner_ob 1.1 03:36:06 13 1 127 9249K 0 6128K 307 1 sleeping #0(1) 0.00000 0.00000 0 993 65 12308 6543 6547790+
740 Brata Agent 1.1 04:48:14 9 1 193 49M 0 15M 712 712 sleeping #1(11) 0.00000 0.00000 501 22184 499 10124 12574348+ 18850962+
654 Brave Browse 0.8 00:25:39 9 1 130 25M 0 14M 578 578 sleeping #0(3) 0.00000 0.00000 501 3884238+ 4257+ 156905+ 51781+ 247913+
627 Brave Browse 0.8 28:43:96 14 4 452 1987M 8160K 225M 578 578 sleeping #1(5) 0.00000 0.00000 501 2903796 920 29604229+ 48511897+ 36407454+
578 Brave Browse 0.5 15:51:36 33 2 1191 449M 0 102M 578 1 sleeping #0(2610) 0.00000 0.00000 501 1569593+ 79698 16757077+ 7385704+ 18539524+
77955 aLacrity 0.5 00:04:69 9 2 223 88M 192K 0 77955 1 sleeping #0(75) 0.00000 0.00000 501 6582 243 4155 293760+ 235108+
80581 Instruments 0.3 02:07:07 8 1 366 2165M 14M 0 80581 1 sleeping #0(181) 0.00000 0.00000 501 832327 1302 239363 82091 10654658+
308 logd 0.2 01:28:76 4 3 1738 9777K 0 3168K 308 1 sleeping #0(1) 0.00000 0.00000 0 70814 70 389959+ 435865+ 2350994+
363 corebrightn 0.2 02:05:54 4 3 130 4081K 0 752K 363 1 sleeping #0(1) 0.08979 0.04316 0 162057+ 96 541205+ 415474+ 1932440+
49581 Brave Browse 0.2 01:11:44 16 1 129 35M+ 0 13M 578 578 sleeping #0(3) 0.00000 0.00000 501 27434+ 729 345891+ 227676+ 462177+
324 powerd 0.1 00:54:27 3 2 135 4497K 0 624K 314 1 sleeping #0(1) 0.00000 0.08322 0 164538+ 105 345765+ 283897+ 1141000+
306 Karabiner-Dr 0.1 00:52:90 8 1 53 6755K 0 4600K 306 1 sleeping #0(1) 0.00500 0.00000 0 29385+ 58 268047+ 101610+ 2833295+
399 airpord 0.1 01:39:20 9 7 147 14M 0 1456K 399 1 sleeping #144(1) 0.00000 0.00000 0 29456 160 818494+ 354979+ 3546410+
402 mDNSResponde 0.0 00:42:43 4 2 204 6148K 0 896K 402 1 sleeping #0(1) 0.00000 0.00000 65 13283 114 164260 78403 1531217+
398 runningboard 0.0 00:41:53 6 5 497 7377K 0 480K 398 1 sleeping #21(1) 0.00000 0.06955 0 26197 90 59487 66443+ 1449632+
4330 GeForceNOWCo 0.0 15:42:45 95 1 284 20M 0 7424K 4330 1 sleeping #1(27) 0.00000 0.00000 501 35667 2813 8470 7712 1075647+
700 karabiner_se 0.0 00:16:27 7 1 41 6993K 0 4800K 700 1 sleeping #0(1) 0.01940 0.00000 0 42872+ 67 56052+ 14179+ 899735+
728 MonitorContr 0.0 00:38:22 5 2 2470 18M 0 5360K 728 1 sleeping #0(681) 0.00000 0.04921 501 11600 223 262856+ 104572+ 384239+
651 Brave Browse 0.0 03:40:10 10 1 163 50M 0 19M 578 578 sleeping #0(3) 0.00000 0.00000 501 175834 831 1925291+ 1536408+ 9574558+
1 launchd 0.0 04:48:05 4 3 2538 22M 0 3376K 1 0 sleeping #0(0) 0.00000 0.03303 0 42465 3409 287306+ 288765+ 1661226+
49821 PerfPowerSer 0.0 00:45:84 4 2 283 12M 256K 2880K 49821 1 sleeping #0(757) 0.00000 0.00000 0 13282 162 908914 612458+ 586205+
825 Brave Browse 0.0 00:17:15 16 1 192 115M 0 28M 578 578 sleeping #0(5) 0.00000 0.00000 501 48091 925 57053+ 21020 228913+
639 sharind 0.0 00:43:78 4 2 245 19M 1408K 6224K 639 1 sleeping #0(1) 0.00000 0.00000 501 15241 274 253887+ 104800 989172+
697 LogiVCoreSe 0.0 00:17:82 8 4 178 11M 0 4988K 697 1 sleeping #0(1) 0.00000 0.00000 501 10678 185 51375+ 3912 676852+
689 karabiner_co 0.0 00:29:28 8 1 103 11M 0 6384K 689 1 sleeping #0(1) 0.01364 0.00000 501 71252+ 130 163648+ 4179+ 967654+
926 Brave Browse 0.0 00:10:61 10 1 220 103M 0 30M 578 578 sleeping #0(4) 0.00000 0.00000 501 29278 879 46940+ 16463 177356+
```

Рис. 2. Перелік процесів та їх властивості у консольній утиліті *top*

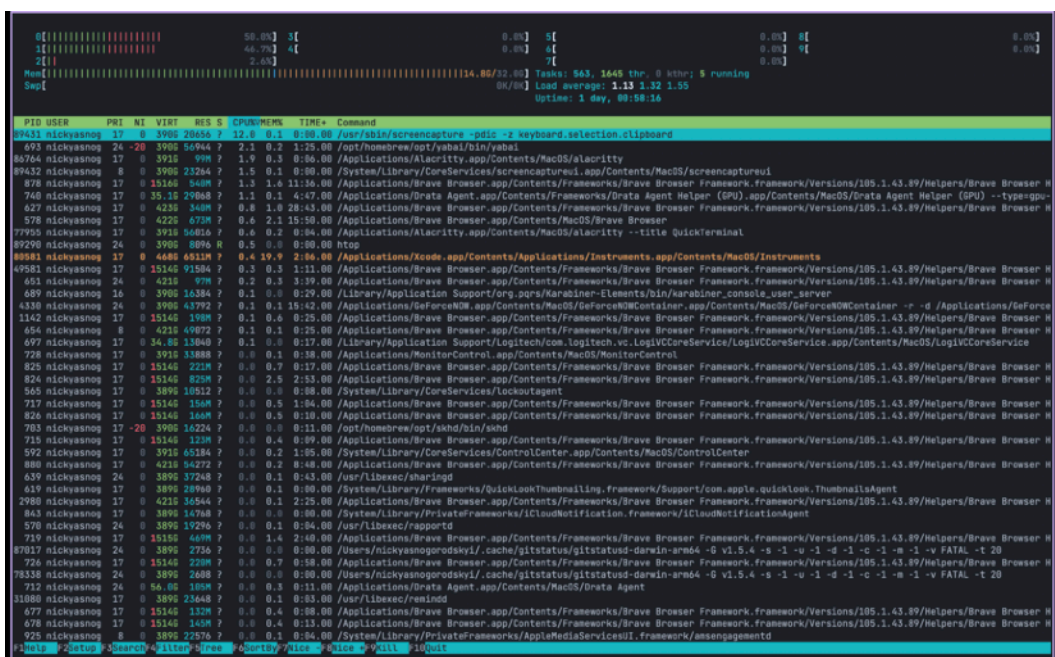


Рис. 3. Перелік процесів та їх властивості у консольній утиліті *htop*

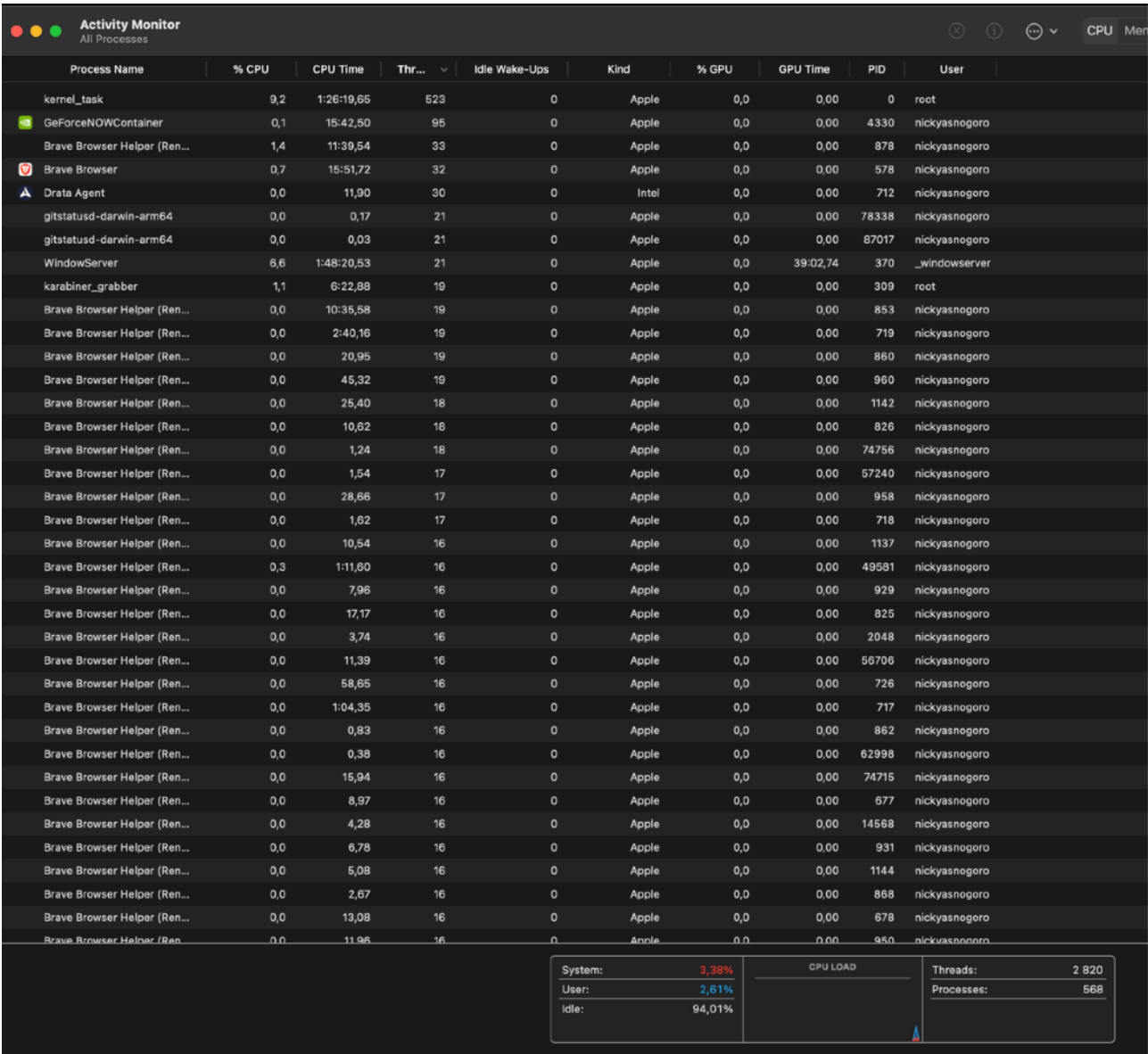


Рис. 4. Перелік процесів та їх властивості у графічному інтерфейсі програми *Activity Monitor*



Використовуючи консольні засоби ОС Linux та сторонні утиліти зміню стан виконання процесу, завершу виконання заданого процесу.

```
Send signal:  PID USER      PRI  NI  VIRT   RES   S  CPU%MEM%  TIME+  Command
0 Cancel      95449 nickyasnog  24   0  389G  1424   ?   0.0  0.0  0:00.00 sleep 1000
1 SIGHUP
2 SIGINT
3 SIGQUIT
4 SIGILL
5 SIGTRAP
6 SIGABRT
6 SIGIOT
7 SIGEMT
8 SIGFPE
9 SIGKILL
10 SIGBUS
```

Рис. 7. Застосування консольної утиліти *htop* для надсилання сигналу до вибраного процесу

```
> pkill sleep
> pgrep -l sleep
94286 sleep
> kill -9 94286
> killall sleep
```

Рис. 8. Застосування консольної утиліти *kill/pkill/pgrep/killall* для надсилання сигналу до вибраного процесу

6. Скомпілюю файл `main.cpp` представлений у лабораторній роботі № 1 і запуску виконуваний файл на різній кількості активних процесорів (ядер). Знайду для даної програми величини  $A$ ,  $S$ ,  $p$  при різних вхідних значеннях величини  $n$ .

```
> taskset -c 0 ./a.out
Set process affinity (cores count) and press <Enter>
Duration: 2176ms
~D <

> pgrep a.out | xargs taskset -c -p
pid 8901's current affinity list: 0
~ >
```

```
> taskset -c 0-1 ./a.out
Set process affinity (cores count) and press <Enter>
Duration: 987ms
~D >

> pgrep a.out | xargs taskset -c -p
pid 9714's current affinity list: 0,1
~ >
```



```
> taskset -c 0-2 ./a.out
Set process affinity (cores count) and press <Enter>
Duration: 728ms
~0 >

> pgrep a.out | xargs taskset -c -p
pid 10034's current affinity list: 0-2
~ >
```

```
> taskset -c 0-3 ./a.out
Set process affinity (cores count) and press <Enter>
Duration: 633ms
~0 >

> pgrep a.out | xargs taskset -c -p
pid 10253's current affinity list: 0-3
~ >
```

```
> taskset -c 0-7 ./a.out
Set process affinity (cores count) and press <Enter>
Duration: 593ms
~0 >

> pgrep a.out | xargs taskset -c -p
pid 10536's current affinity list: 0-7
~ >
```

Після 4-ох ядер пришвидшення майже зупиняється

$$A = \frac{2176}{633} = 3.43$$

$$3.42 = \frac{1}{p + \frac{p-1}{4}} \quad 3.42 = \frac{4}{5p-1} \quad 5p-1 = 1.169 \quad 5p = 2.169 \quad p = 0.43$$

Приблизно 43% програмного коду виконується послідовно.

7. Слідуючи за індивідуальним завданням, досліджу компілювання c++ за допомогою команди **g++**.

Вхідні дані: програма з минулого пункту.

```
> time g++ main.cpp -pthread  
g++ main.cpp -pthread 0.83s user 0.10s system 99% cpu 0.932 total
```

Процес зайняв 0.932 секунд. Зміна к-сті ядер не впливає на швидкість виконання (певно весь код послідовний)

### **Висновки**

Я ознайомився з поняттями процес та потік в операційних системах Linux та MacOS, зрозумів що основні характеристики однакові. Дізнався більше про консольні застосунки для керування процесами, поверхово зрозумів різницю між Linux та BSD