

Laboratorio # 1 Sistemas Operativos Grupo 80

**2242232-2724, Orejuela Albornoz
2340816-2724, Albornoz Suárez
2343025-2724, Silva Zuñiga**

**Operating Systems
Jefferson A.
Cali, 2024-II**

Laboratorio # 1 Sistemas Operativos Gr. 80.

General Description

The following outlines the guidelines for a theoretical-practical activity for students in the Operating Systems course. The activity specifically assesses their mastery of some basic concepts, definitions related to CPU management, processes, scheduling, and the implementation of a simulator using C or C++.

Objectives During the development of the activities, students will achieve the following:

- Enumerate the milestones in the evolution of computer systems.
- Describe the objectives and functions of modern operating systems.
- Simulate the behavior of the CPU and process management.
- Apply the principles of scheduling.

Before You Begin

Read the following:

- *Operating System Concepts* (9th Edition) by Silberschatz and Galvin: Chapters 1, 2, and 3
- *Modern Operating Systems* by Tanenbaum: Chapters 1 and 2
-

Create a git repository and add the link:

Use this document as template [here]

Activity No. 1: Conceptualization [40%]

After completing the recommended readings in this document, answer the following questions in your own words, concisely and accurately.

1. Enumerate the milestones in the evolution of computer systems

HISTORY OF OPERATING SYSTEMS

The First Generation (1945–55): Vacuum Tubes

These early computers were primitive, using vacuum tubes or relays, and required direct programming in machine language or by wiring plugboards. By the early 1950s, the process improved slightly with the introduction of punched cards for programming.

The Second Generation (1955–65): Transistors and Batch Systems

The introduction of the transistor in the mid-1950s revolutionized computing by making computers more reliable and allowing them to be sold to customers.

The Third Generation (1965–1980): ICs and Multiprogramming

In the early 1960s, IBM introduced the System/360, a series of compatible computers for both scientific and commercial use, simplifying product lines and ensuring program compatibility across models. This approach continued with the 370 and zSeries.

The Fourth Generation (1980–Present): Personal Computers

The development of LSI circuits led to personal computers. Intel's 8080 microprocessor in 1974 used CP/M, an OS developed by Gary Kildall. IBM missed out on CP/M and instead bought DOS from Seattle Computer Products through Bill Gates. MS-DOS became dominant, evolving with features from UNIX.

The Fifth Generation (1990–Present): Mobile Computers

The idea of portable communication devices has been around since the 1940s, starting with large mobile phones and evolving to the first handheld "brick" phones in the 1970s. Today, mobile phones are ubiquitous, used for much more than calling.

2. What are the four components of a computer system? Describe each one.

Processors

The "brain" of the computer is the CPU. It fetches instructions from memory and executes them. The basic cycle of every CPU is to fetch the first instruction from memory, decode it to determine its type and operands, execute it, and then fetch, decode, and execute subsequent instructions. The cycle is repeated until the program finishes. In this way, programs are carried out.

Memory

The second major component in any computer is the memory. Ideally, a memory should be extremely fast (faster than executing an instruction so that the CPU is not held up by the memory), abundantly large, and dirt cheap. No current technology satisfies all of these goals, so a different approach is taken. The memory system is constructed as a hierarchy of layers. The top layers have higher speed, smaller capacity, and greater cost per bit than the lower ones, often by factors of a billion or more.

Disk

Next in the hierarchy is magnetic disk (hard disk). Disk storage is two orders of magnitude cheaper than RAM per bit and often two orders of magnitude larger as well. The only problem is that the time to randomly access data on it is close to three orders of magnitude slower. The reason is that a disk is a mechanical device,

I/O Devices

These devices consist of a controller and the device itself. The controller, often a chip or set of chips, handles commands from the operating system, like reading data, and performs complex tasks such as converting data formats and managing device operations. The actual device has a simpler interface but relies on the controller to interact with the operating system. Device drivers are specialized software needed to manage each type of controller and are typically integrated into the operating system kernel, though some systems allow drivers to run in user space.

3. What is the difference between a monolithic kernel and a microkernel?

In a monolithic kernel, the entire operating system operates as a single program in kernel mode. This approach allows for efficient inter-procedure calls but can lead to a complex and potentially unstable system, as a bug in any part of the kernel can crash the entire system. The system is organized into a main program, service procedures, and utility procedures, all of which run in kernel mode.

In contrast, a microkernel architecture splits the operating system into a small kernel that runs in kernel mode and many other modules (like device drivers and file systems) that run in user mode. This separation improves system reliability because a bug in one of these user-mode components does not affect the entire system. The microkernel handles essential functions such as interprocess communication and basic scheduling, while other functionalities are managed by separate user processes. This design makes the system more modular and resilient to crashes.

4. Define an Operating System from two different perspectives.

An operating system (OS) serves two main functions: abstracting hardware complexity and managing hardware resources.

Operating System as an Extended Machine:

The OS provides an abstraction layer that simplifies the interaction with hardware. For example, it hides the complexities of hardware like SATA disks behind simpler interfaces like file systems, enabling programs to perform tasks (e.g., reading/writing files) without dealing with intricate hardware details. This abstraction makes complex hardware interactions manageable and efficient.

Operating System as a Resource Manager:

The OS manages and allocates system resources such as CPU, memory, and I/O devices among multiple programs and users. It ensures orderly access to these resources to prevent conflicts and inefficiencies. For instance, it handles tasks like time and space multiplexing—time multiplexing allows multiple programs to share a single resource like the

CPU or printer in turns, while space multiplexing involves sharing resources like memory and disk space simultaneously among different programs.

5. What is the purpose of system calls?

Answer/ System calls act as a bridge between user programs and the operating system, allowing programs to access services and resources that are typically managed by the operating system. When a program needs to perform a task that requires system-level permissions, it makes a system call, triggering a special instruction that shifts control to the operating system's kernel. The kernel, which operates in a secure and privileged mode, takes over, interprets the request, checks the parameters, and carries out the required operation. Once the task is completed, control is handed back to the user program.

6. What is a multiprogrammed operating system?

Answer// Is a type of operating system designed to maximize CPU utilization by allowing multiple programs to be loaded and executed concurrently.

Four main characteristics:

1. **Concurrency:** several processes are kept in memory at the same time, allowing the operating system to switch between processes and execute them in an interleaved manner, which helps in efficiently utilizing the CPU.
2. **Process Management:** using techniques like scheduling and context switching, processes are selected for execution based on a scheduling algorithm, and the state of a process is saved and restored during context switches.
3. **Memory Management:** using memory management techniques such as partitioning (dividing memory into sections) or paging (dividing memory into fixed-size pages) to allocate memory to multiple processes.
4. **I/O and CPU Overlap:** When one process is waiting for I/O operations (like reading from a disk), the CPU can be allocated to another process that is ready to execute. This overlap of I/O operations and CPU execution helps in maintaining high CPU utilization.

7. What is a process?

Answer/ A process is a program that has been loaded into memory and is currently being executed by the CPU. In other words a process could be considered as a running instance.

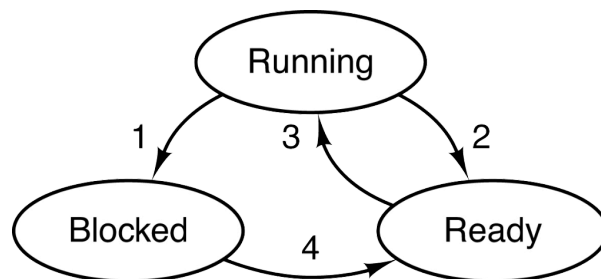
8. What are the states of a process?

Answer / “each process is an independent entity, with its own program counter and internal state, processes often need to interact with other processes”

Andrew S. Tanenbaum. Modern operating systems . pearson, 1992.

The state of a process is determined by current activity of the process. According to Tanunbaum there is three states that a process can be:

- Running (actually using the CPU at that instant).
- Ready (runnable; temporarily stopped to let another process run).
- Blocked (unable to run until some external event happens).



1. Process blocks for input
2. Scheduler picks another process
3. Scheduler picks this process
4. Input becomes available

image from “Andrew S. Tanenbaum. Modern operating systems . pearson, 1992.”

9. What information is stored in the Process Control Block (PCB) associated with a process?

Answer/ The PCB stored process tables, it means arrays of structures with information like process state, process ID, program counter, cpu registers, memory management information, scheduling information, accounting information, i/o status information, list of open files and process hierarchy information.

10. What are the main activities of an operating system in relation to process management?

Answer/ In the context of process management, here are the main activities:

1. Scheduling processes and threads on the cpus
2. Creating and deleting both user and system processes
3. Suspending and resuming processes
4. Providing mechanisms for process synchronization
5. Providing mechanisms for process communication.

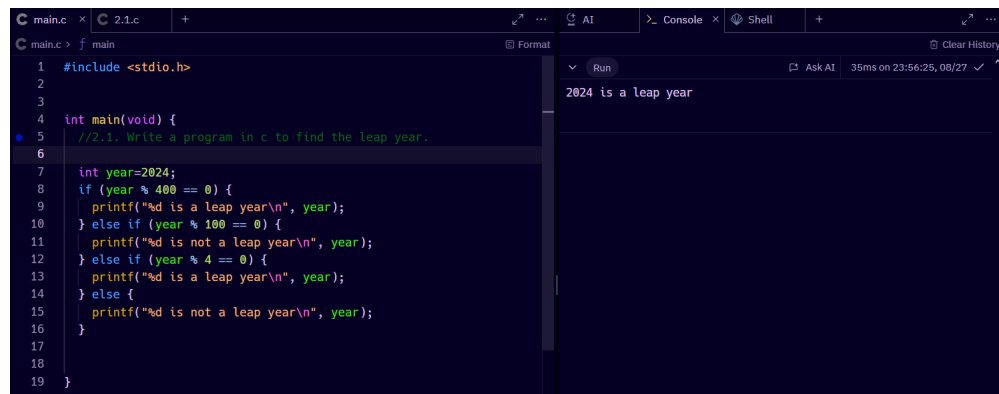
Activity No. 2: Remembering the C and C++ languages [50%]

1. Implements the programs and functions below:

1 Write a program in c to find the leap year.

```
#include <stdio.h>
```

```
int main(void) {  
    //2.1. Write a program in c to find the leap year.  
  
    int year=1900;  
    if (year % 400 == 0) {  
        printf("%d is a leap year\n", year);  
    } else if (year % 100 == 0) {  
        printf("%d is not a leap year\n", year);  
    } else if (year % 4 == 0) {  
        printf("%d is a leap year\n", year);  
    } else {  
        printf("%d is not a leap year\n", year);  
    }  
}
```



The screenshot shows a code editor with a C program. The code defines a variable 'year' as 2024 and uses a series of if-else statements to check for leap years based on divisibility by 400, 100, and 4. The output window on the right shows the result: '2024 is a leap year'.

```
C main.c x C 2.1.c +  
C main.c > f main  
1 #include <stdio.h>  
2  
3  
4 int main(void) {  
5     //2.1. Write a program in c to find the leap year.  
6  
7     int year=2024;  
8     if (year % 400 == 0) {  
9         printf("%d is a leap year\n", year);  
10    } else if (year % 100 == 0) {  
11        printf("%d is not a leap year\n", year);  
12    } else if (year % 4 == 0) {  
13        printf("%d is a leap year\n", year);  
14    } else {  
15        printf("%d is not a leap year\n", year);  
16    }  
17  
18  
19 }
```

Run 35ms on 23:56:25, 08/27
2024 is a leap year

here declaring the variable "year" as 1900(is not a leap year)

```
C main.c x C 2.1.c +
C main.c > f main
1 #include <stdio.h>
2
3
4 int main(void) {
5     //2.1. Write a program in c to find the leap year.
6
7     int year=1900;
8     if (year % 400 == 0) {
9         printf("%d is a leap year\n", year);
10    } else if (year % 100 == 0) {
11        printf("%d is not a leap year\n", year);
12    } else if (year % 4 == 0) {
13        printf("%d is a leap year\n", year);
14    } else {
15        printf("%d is not a leap year\n", year);
16    }
17
18
19 }
```

Run 29ms on 23:57:18, 08/27 ✓

1900 is not a leap year

2

Write a program in c to calculate factorial of a number

```
#include <stdio.h>
```

```
int main() {
```

```
    int number = 3; // Example input
```

```
    int acum = 1; // Initialize the accumulator to 1
```

```
    // Factorial calculation directly in main
```

```
    if (number == 0) {
```

```
        acum = 1; // Base case: 0! = 1
```

```
    } else {
```

```
        for (int i = 1; i <= number; i++) {
```

```
            acum = acum * i; // Multiply acum by each number from 1 to
```

```
'number'
```

```
        }
```

```
    }
```

```
    // Print the result
```

```
    printf("The factorial of %d is %d\n", number, acum);
```

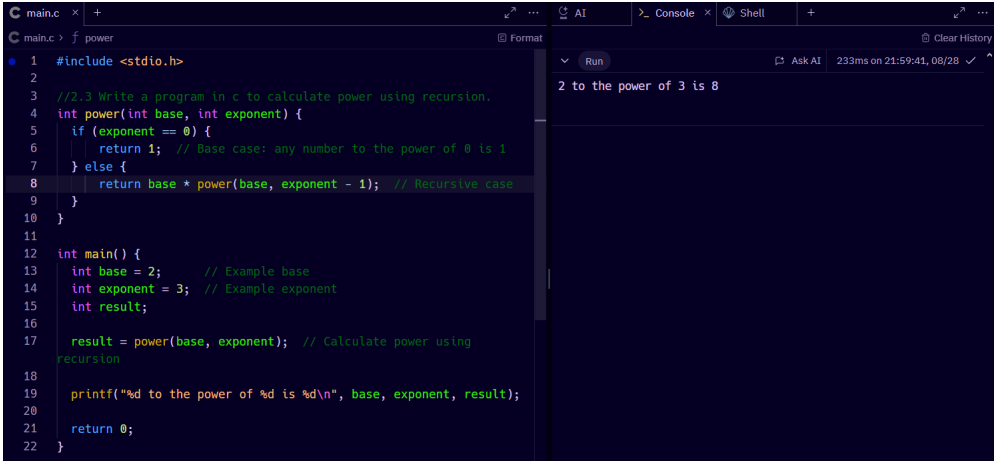
```
    return 0;
```

```
}
```

```
C main.c x C 2.1.c +
C main.c > f main
1 #include <stdio.h>
2
3 int main() {
4     int number = 3; // Example input
5     int acum = 1; // Initialize the accumulator to 1
6
7     // Factorial calculation directly in main
8     if (number == 0) {
9         acum = 1; // Base case: 0! = 1
10    } else {
11        for (int i = 1; i <= number; i++) {
12            acum = acum * i; // Multiply acum by each number from
13            1 to 'number'
14        }
15    }
16
17    // Print the result
18    printf("The factorial of %d is %d\n", number, acum);
19
20    return 0;
21 }
```

Run 21ms on 00:00:45, 08/28 ✓

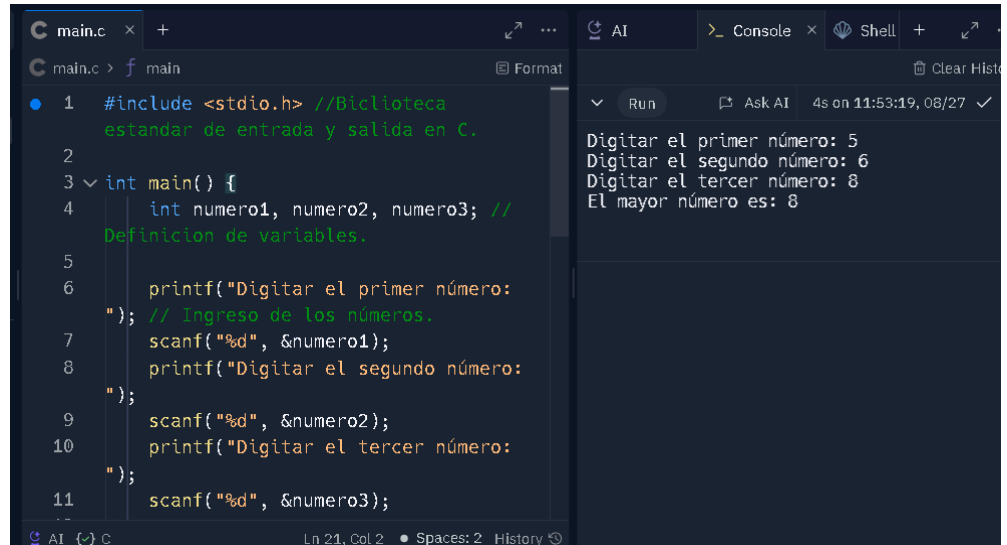
The factorial of 3 is 6

3	<p>Write a program in c to calculate power using recursion.</p> <pre>#include <stdio.h> //2.3 Write a program in c to calculate power using recursion. int power(int base, int exponent) { if (exponent == 0) { return 1; // Base case: any number to the power of 0 is 1 } else { return base * power(base, exponent - 1); // Recursive case } } int main() { int base = 2; // Example base int exponent = 3; // Example exponent int result; result = power(base, exponent); // Calculate power using recursion printf("%d to the power of %d is %d\n", base, exponent, result); return 0; }</pre> 
4	<p>Write a program in c to find even or odd numbers.</p> <pre>#include <stdio.h></pre>

	<pre> int main() { int number; printf("Ingrese un número entero. "); scanf("%d", &number); if (number % 2 == 0) { printf("%d es par ", number); } else{ printf("%d es impar ", number); } return 0; } </pre>
5	<p>Write a program in c to print fabonnaci series. #include <stdio.h></p> <pre> int main() { int n, primero = 0, segundo = 1, siguiente; printf("Digite el numero de terminos: "); scanf("%d", &n); printf("La serie de Fibonacci: \n"); for (int i = 0; i < n; i++) { if (i <= 1){ siguiente = i; } else { siguiente = primero + segundo; primero = segundo; segundo = siguiente; } printf("%d\n", siguiente); } return 0; } </pre>
6	<p>Write a function to check uppercase letter #include <stdio.h> #include <ctype.h></p>

	<pre> int esMayuscula(char caracter) { // Verifica si en la tabla ASCII la variable esta entre los valroes de las mayusculas if (caracter >= 'A' && caracter <= 'Z') { return 1; // si es mayúscula retornará uno } else { return 0; } } int main() { char caracter; printf("Ingrese el caracter:"); scanf("%c", &caracter); // se llama a la funcion if (esMayuscula(caracter)) { printf("%c es una letra mayúscula. \n", caracter); }else{ printf("%c no es una letra mayúscula. \n", caracter); } return 0; } </pre>
7	<p>Write a program in c to function to check lowercase letter.</p> <pre> #include <stdio.h> #include <ctype.h> int esMinuscula(char caracter) { // Verifica si en la tabla ASCII la variable esta entre los valroes de las minusculas if (caracter >= 'a' && caracter <= 'z') { return 1; // si es minuscula retornará uno } else { return 0; } } </pre>

	<pre> } int main() { char caracter; printf("Ingrese el caracter:"); scanf("%c", &caracter); // se llama a la funcion if (esMinuscula(caracter)) { printf("%c es una letra minuscula. \n", caracter); }else{ printf("%c no es una letra minuscula. \n", caracter); } return 0; } </pre>
8	<p>Find the greater of the three numbers.</p> <pre> #include <stdio.h> //Biblioteca estándar de entrada y salida en C. int main() { int numero1, numero2, numero3; // Definición de variables. printf("Digitar el primer número: "); // Ingreso de los números. scanf("%d", &numero1); printf("Digitar el segundo número: "); scanf("%d", &numero2); printf("Digitar el tercer número: "); scanf("%d", &numero3); if(numero1 >= numero2 && numero1 >= numero3){ //Estructura condicional para determinar el numero mayor printf("El mayor número es: %d\n", numero1); } else if(numero2 >= numero1 && numero2 >= numero3){ printf("El mayor número es: %d\n", numero2); } else{ printf("El mayor número es: %d\n", numero3); } return 0; } </pre>



```
1 #include <stdio.h> //Biblioteca
   estandar de entrada y salida en C.
2
3 int main() {
4     int numero1, numero2, numero3; //
   Definicion de variables.
5
6     printf("Digitar el primer número:
   "); // Ingreso de los números.
7     scanf("%d", &numero1);
8     printf("Digitar el segundo número:
   ");
9     scanf("%d", &numero2);
10    printf("Digitar el tercer número:
   ");
11    scanf("%d", &numero3);
```

Console Output:

```
Digitar el primer número: 5
Digitar el segundo número: 6
Digitar el tercer número: 8
El mayor número es: 8
```

9 Write a program in c to type casting implicit explicit.

//Biblioteca estándar para ingreso y salida de datos
#include <stdio.h>
//Definicion de variables a transformar
int main() {
 int intVar;
 float floatVar;
 double doubleVar;

 //Ingreso número entero
 printf("Ingrese un número entero: ");
 scanf("%d", &intVar);

 //Ingreso dato tipo float
 printf("Ingrese un número tipo float: ");
 scanf("%f", &floatVar);

 //Conversión implícita de int a double
 doubleVar= intVar;
 printf("Conversión implícita (int a double): %d a %.2f\n", intVar,
doubleVar);

 //Conversión implícita de float a double
 doubleVar= floatVar;
 printf("Conversión implícita (float a double): %.2f a %.2f\n", floatVar,
doubleVar);

 //Conversión explícita de float a int
 intVar= (int)floatVar;
 printf("Conversión explícita (float a int): %.2f a %d\n", floatVar, intVar);

 return 0;
}

```
main.c x +
C main > f main
Format
1 //Biblioteca estandar para ingreso y salida de datos
2 #include <stdio.h>
3 //Definición de variables a transformar
4 int main() {
5     int intVar;
6     float floatVar;
7     double doubleVar;
8
9     //Ingreso número entero
10    printf("Ingrese un número entero: ");
11    scanf("%d", &intVar);
12
13    //Ingreso dato tipo float
14    printf("Ingrese un número tipo float: ");
15    scanf("%f", &floatVar);
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
```

2. Write a C program that receives and processes grades for the Operating Systems (SO) course using structures.

```
// Biblioteca estándar de entrada y salida en C
#include <stdio.h>
// Estructura para almacenar la información del estudiante
struct Estudiante {
    char nombre[30];
    int nota;
};

int main() {
    int cantidad, i;
    float promedio = 0.0;
    int notaMaxima = -1; //Variable inicializada con un valor menor que cualquier calificación posible como referencia.
    int notaMinima = 6; //Variable inicializada con un valor mayor que cualquier calificación posible como referencia.

    //Solicitar al usuario el número de estudiantes
    printf("Ingrese el número de estudiantes en el curso: ");
    scanf("%d", &cantidad);

    // Array de estructuras Estudiante
    struct Estudiante estudiantes[cantidad];

    // Ciclo for para ingresar información de los estudiantes
    for(i = 0; i < cantidad; i++) {
        printf("Ingrese el nombre del estudiante %d: ", i + 1);
        scanf("%s", estudiantes[i].nombre);
        printf("Ingrese la calificación de %s: ", estudiantes[i].nombre);
        scanf("%d", &estudiantes[i].nota);

        //Suma de calificaciones para el promedio
        promedio += estudiantes[i].nota;

        //Calificación más alta
        if(estudiantes[i].nota > notaMaxima) {
            notaMaxima = estudiantes[i].nota;
        }

        //Calificación más baja
        if(estudiantes[i].nota < notaMinima) {
            notaMinima = estudiantes[i].nota;
        }
    }

    //Cálculo del promedio de las calificaciones
    promedio /= cantidad;

    //Resultados
```

```

printf("\nReporte de calificaciones:\n");
printf("Promedio de calificaciones: %.2f\n", promedio);
//Ciclo for para mostrar las calificaciones de los estudiantes
for (i = 0; i < cantidad; i++) {
    printf("Estudiante: %s, Calificación: %d\n", estudiantes[i].nombre,
estudiantes[i].nota);
}
printf("Calificación más alta: %d\n", notaMaxima);
printf("Calificación más baja: %d\n", notaMinima);
return 0;
}

```

The screenshot shows a C++ IDE with a code editor on the left and a console output on the right. The code in the editor defines a struct 'Estudiante' with fields 'nombre' and 'nota', and a 'main' function that takes the number of students, their names, and grades as input. The console output shows the program's execution: it asks for the number of students (2), then for each student's name and grade (Alex with grade 5, and Alexander with grade 1). Finally, it displays the report: average grade 3.00, highest grade 5, and lowest grade 1.

3. Write a C++ program that performs the following tasks: First, the program should receive a numeric value and determine whether it is a prime number, displaying the result. Second, the program should accept a list of numbers and identify the prime numbers within that list. Lastly, the program should allow the user to input a numeric range and display all prime numbers within that specified range.

Answer: `#include <iostream>`

`#include <vector>`

`/*Write a C++ program that performs the following tasks: First, the program should receive a numeric value and determine whether it is a prime number, displaying the result. Second, the program should accept a list of numbers and identify the prime numbers within that list. Lastly, the program should allow the user to input a numeric range and display all prime numbers within that specified range.*/`

`// Function to check if a number is prime`

`bool isPrime(int num) {`


```

    if (num <= 1) return false;

    for (int i = 2; i * i <= num; ++i) {
        if (num % i == 0) return false;
    }

    return true;
}

```

// Function to display primes in a list or range

```

void displayPrimes(const std::vector<int>& numbers) {

    std::cout << "Prime numbers: ";

    for (int num : numbers) {
        if (isPrime(num)) {
            std::cout << num << " ";
        }
    }

    std::cout << "\n";
}

```

```

int main() {

    int choice;

    std::cout << "Select an option:\n";

    std::cout << "1. Check if a single number is prime\n";

    std::cout << "2. Identify prime numbers in a list\n";

    std::cout << "3. Display prime numbers within a range\n";

    std::cout << "Enter your choice: ";
}

```

```
std::cin >> choice;

if (choice == 1) {
    int num;

    std::cout << "Enter a number: ";

    std::cin >> num;

    if (isPrime(num)) {
        std::cout << num << " is a prime number.\n";
    } else {
        std::cout << num << " is not a prime number.\n";
    }
} else if (choice == 2) {
    int n;

    std::cout << "Enter the number of elements: ";

    std::cin >> n;

    std::vector<int> numbers(n);

    std::cout << "Enter the numbers:\n";

    for (int i = 0; i < n; ++i) {
        std::cin >> numbers[i];
    }

    displayPrimes(numbers);
} else if (choice == 3) {
    int start, end;

    std::cout << "Enter the start of the range: ";

    std::cin >> start;

    std::cout << "Enter the end of the range: ";
```

```

std::cin >> end;

std::vector<int> numbers;

for (int i = start; i <= end; ++i) {

    numbers.push_back(i);

}

displayPrimes(numbers);

} else {

    std::cout << "Invalid choice.\n";

}


return 0;

}

```

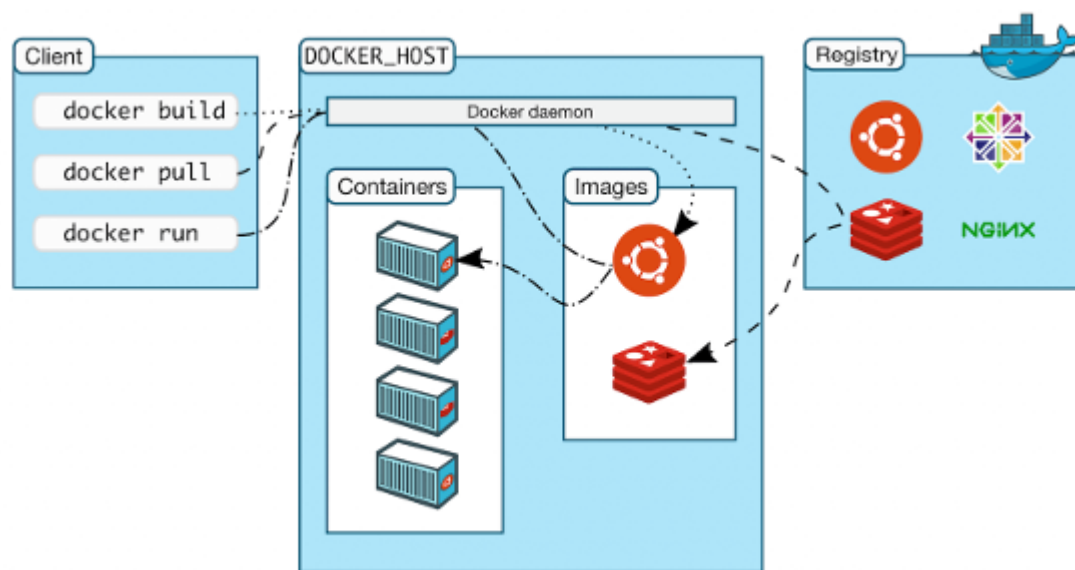
4. Create a program with C++ to model geometric shapes and perform area, perimeter, and color operations. Rectangle, square, triangle, etc. shapes must be considered. Tip. Use classes and inheritance. //this program is in folder 4 of activity 2

Activity No. 3: Docker practical session [10%]

Docker is a command line-based software allowing users to manipulate images and create application containers.

As presented in the course, Docker consists of two elements:

- a client, to receive commands from the user
- a server, to execute commands and manage images and containers



Docker commands architecture

Typing this command will give the Client and Server versions available on your computer.

Paste a screenshot of result

```
docker version
```

```
invitado@s2pc11:~$ docker version
Client: Docker Engine - Community
 Cloud integration: v1.0.35+desktop.5
 Version:          25.0.3
 API version:      1.44
 Go version:       go1.21.6
 Git commit:       4debf41
 Built:            Tue Feb  6 21:13:09 2024
 OS/Arch:          linux/amd64
 Context:          default

Server: Docker Engine - Community
 Engine:
  Version:          25.0.3
  API version:      1.44 (minimum version 1.24)
  Go version:       go1.21.6
  Git commit:       f417435
  Built:            Tue Feb  6 21:13:09 2024
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.6.28
  GitCommit:        ae07eda36dd25f8a1b98dfbf587313b99c0190bb
 runc:
  Version:          1.1.12
  GitCommit:        v1.1.12-0-g51d5e94
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
```

Usage, options and a full list of available commands can be accessed through the command line in a terminal. Type the following command

```
docker --help
```

The general usage of a Docker command line is as follows:

```
docker [OPTIONS] COMMAND [arg...]
```

Questions

1. How many arguments are absolutely required by the command 'docker pull' ?

Answer/

According to docker docs there's two ways to write the command:

This command is for download an image from a registry

Usage: `docker image pull [OPTIONS] NAME[:TAG|@DIGEST]`

Aliases:

```
docker pull
```

Docker.Docs. <https://docs.docker.com/reference/cli/docker/image/pull/>.

We will focus on the shorthand of this command “docker pull”

and his basic syntax is: **docker pull [OPTIONS] NAME[:TAG|@DIGEST]**.

NAME[:TAG|@DIGEST] (required):

- **NAME**: The name of the image to pull.
- **TAG**: An optional tag to specify a version of the image. If omitted, Docker defaults to **latest**.
- **DIGEST**: An optional digest for a specific version of the image.

Options (optional):

- **--all-tags, -a**: Pull all tags of an image.
- **--platform**: Specify the platform to pull the image for (e.g., **linux/amd64**, **linux/arm64**).

Answering the main question the command primarily takes a single argument which is the name of the image that is needed to pull. Add an optional arguments are helpful to refine the pull operation.

2. Do you remember what a registry is?

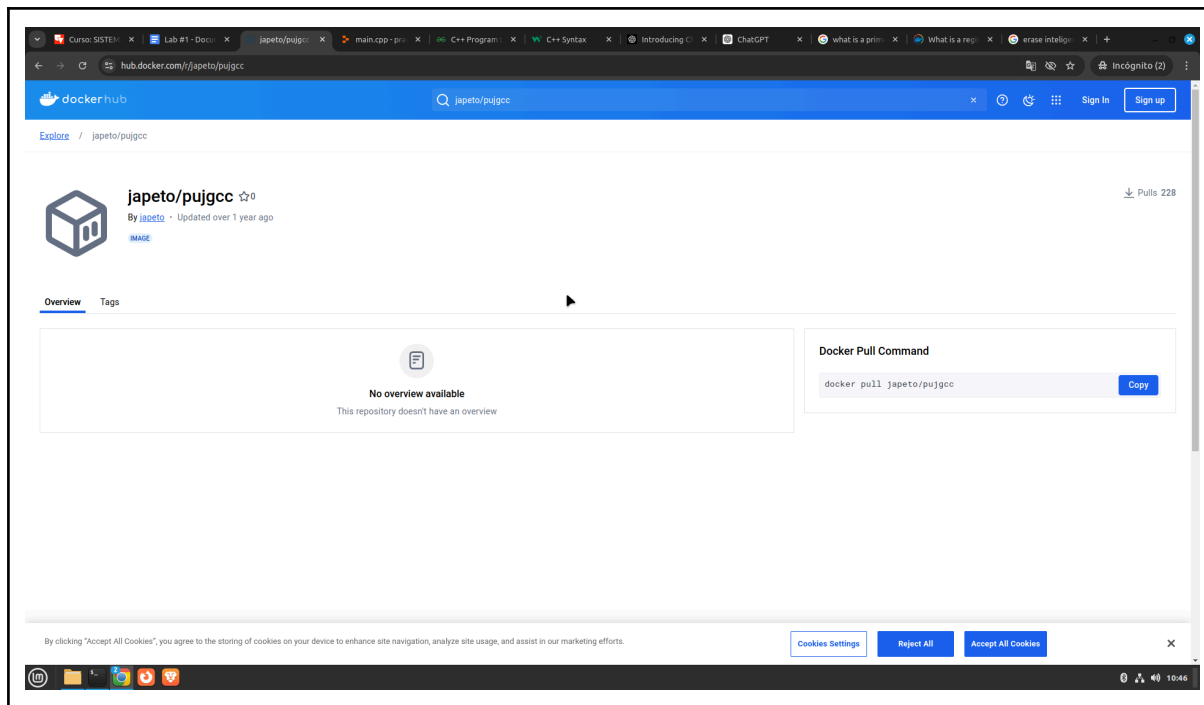
Answer/

Yes, I do, an image registry is a location that hosts storing and sharing container images. In this case the registry by default is Docker hub, this register can be public or private.

Download a predefined image available on the DockerHub

In a web browser, navigate to the DockerHub : <https://hub.docker.com/>

In the top search bar, type : japeto/pujgcc and paste a screenshot



Our course has images available for the development of practical sessions.

Questions

1. How many times was the *japeto* image downloaded ?

Answer/ in the moment shows that was pull **228 times and 139 downloads**.

Execute the command inside a terminal.

Paste a screenshot of result

```
docker pull japeto/pujgcc:v0.12
```

```
invitado@s2pc11:~$ docker pull japeto/pujgcc:v0.12
Error response from daemon: manifest for japeto/pujgcc:v0.12 not found: manifest unknown: manifest unknown
invitado@s2pc11:~$
```

You will get an error as this image has no default tag ("latest"). So we need to specify one in the command line.

Go to the "Tags" tab and copy the pull command of version latest

Paste a screenshot of result

docker pull japeto/pujgcc:v0.12

```
invitado@s2pc11:~$ docker pull japeto/pujgcc:latest
latest: Pulling from japeto/pujgcc
aa18ad1a0d33: Pull complete
15a33158a136: Pull complete
f67323742a64: Pull complete
c4b45e832c38: Pull complete
e5d4afe2cf59: Pull complete
1efbd2d5674a: Pull complete
022a58c161f7: Pull complete
6461d3be7619: Pull complete
8287aab02fcd: Pull complete
edd3d1d6fad6: Pull complete
c6dbbe32db7e: Pull complete
Digest: sha256:9b3d7bbf410396d0a26e6bcffbb54e4239736d5a23ad694b03d93c26f9fc6868
Status: Downloaded newer image for japeto/pujgcc:latest
docker.io/japeto/pujgcc:latest

What's Next?
 1. Sign in to your Docker account → docker login
 2. View a summary of image vulnerabilities and recommendations → docker scout quickview japeto/pujgcc:latest
invitado@s2pc11:~$
```

Question:

1. How many times do you see 'Pull complete' displayed ? Why ?

Answer/ is displayed 11 times, pull complete appears many times, because the images are made by layers, when an image is pulled each layer is pulled separately. The message "Pull complete" indicates that a specific layer has been fully downloaded.

Now, to be sure that the image was correctly pulled, let's see the list of all available downloaded images inside our workspace.

Paste a screenshot of result

docker image


```

invitado@s2pc11:~$ docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
nginx               latest     a72860cb95fd  2 months ago  188MB
dockercoins-hashter latest     01c4c2462c38  2 months ago  311MB
dockercoins-webui   latest     5630d72d47d2  2 months ago  221MB
dockercoins-worker   latest     fea41ee4b9a1  2 months ago  67.7MB
dockercoins-rng      latest     370add9542b6  2 months ago  66.9MB
dpage/pgadmin4       latest     c3d1e82c1e18  3 months ago  485MB
redis               latest     1a83fd5edeed  3 months ago  117MB
postgres            12         feaa4007f007  3 months ago  419MB
postgres            <none>     6f2bfffbe39bf 3 months ago  434MB
gcr.io/k8s-minikube/kicbase v0.0.44    5a6e59a9bdc0  3 months ago  1.26GB
dpage/pgadmin4       <none>     5675b83f2460  4 months ago  502MB
dpage/pgadmin4       <none>     f3e7a186b202  4 months ago  476MB
dpage/pgadmin4       <none>     3158f8135bcd  5 months ago  472MB
postgres            <none>     766c77e81904  6 months ago  419MB
postgres            <none>     1d9988eca73d  6 months ago  419MB
postgres            <none>     735b07d34bf1  6 months ago  419MB
dpage/pgadmin4       <none>     a0786aa69feb  6 months ago  471MB
alpine              latest     05455a08881e  7 months ago  7.38MB
bash                latest     f208c2c858b1  8 months ago  14MB
jupyter/datascience-notebook latest     f78a42f3bc9a  10 months ago 5.92GB
japeto/pujgcc        latest     05141310a94c  18 months ago 1.39GB
invitado@s2pc11:~$

```

Question:

1. What is the size of the japeto/pujgcc image ?
Answer/ Japeto/pujgcc size is 1.39 GB

Perform a task using a pulled image

Among the Docker commands, we will now use the 'run' command.

Question:

1. What are the options and parameters of the 'run' command ?

Answer/ Japeto

```
docker run --help
```

As displayed in the terminal, the description of the command is 'Run a command in a new container'.

Question :

1. What is the difference between an image and a container ?

Answer

Now, to run the application, execute the following command:

Paste a screenshot of result

```
docker run japeto/pujgcc bash --help
```

```
invitado@s2pc11:~$ docker run japeto/pujgcc bash --help
GNU bash, version 4.3.30(1)-release-(x86_64-pc-linux-gnu)
Usage: bash [GNU long option] [option] ...
       bash [GNU long option] [option] script-file ...
GNU long options:
  --debug
  --debugger
  --dump-po-strings
  --dump-strings
  --help
  --init-file
  --login
  --noediting
  --noprofile
  --norc
  --posix
  --rcfile
  --restricted
  --verbose
  --version
Shell options:
  -ilrsD or -c command or -O shopt_option          (invocation only)
  -abefhkmnptuvxBCHP or -o option
Type `bash -c "help set"' for more information about shell options.
Type `bash -c help' for more information about shell builtin commands.
Use the `bashbug' command to report bugs.
invitado@s2pc11:~$
```

Congratulations!

You just successfully downloaded and used your first Docker image!

Running *PUJGCC* without parameters was interesting as a demonstration of Docker's features. But if we want to really run *PUJGCC*, we also need to provide parameters and, most importantly, input files.

Find the paths to bind

To bind our current folder to the `/data/` folder located inside a container, we first need the absolute path of the current folder, obtained through the `unix pwd` command.

Paste a screenshot of result

```
pwd
```

```
invitado@s2pc11:~$ pwd
/home/invitado
invitado@s2pc11:~$
```

This path will be used in further commands through `${PWD}`.

Instead of running `ls` command to `/home/` files, we will now just list the content of the `/data/` folder inside the container but bind with the host.

Paste a screenshot of result

```
docker run japeto/pujgcc:latest ls /data
```

```
invitado@s2pc11:~$ docker run japeto/pujgcc:latest ls /data
ls: cannot access /data: No such file or directory
invitado@s2pc11:~$
```

If nothing appears, it is normal: the folder is empty and only serves as a “*branching point*”.

We now have the paths of the two folders we want to bind together.

Bind a local folder into a container

To perform the folder mapping between the current folder and `/data` inside the image, the syntax is simple. *Paste a screenshot of result*

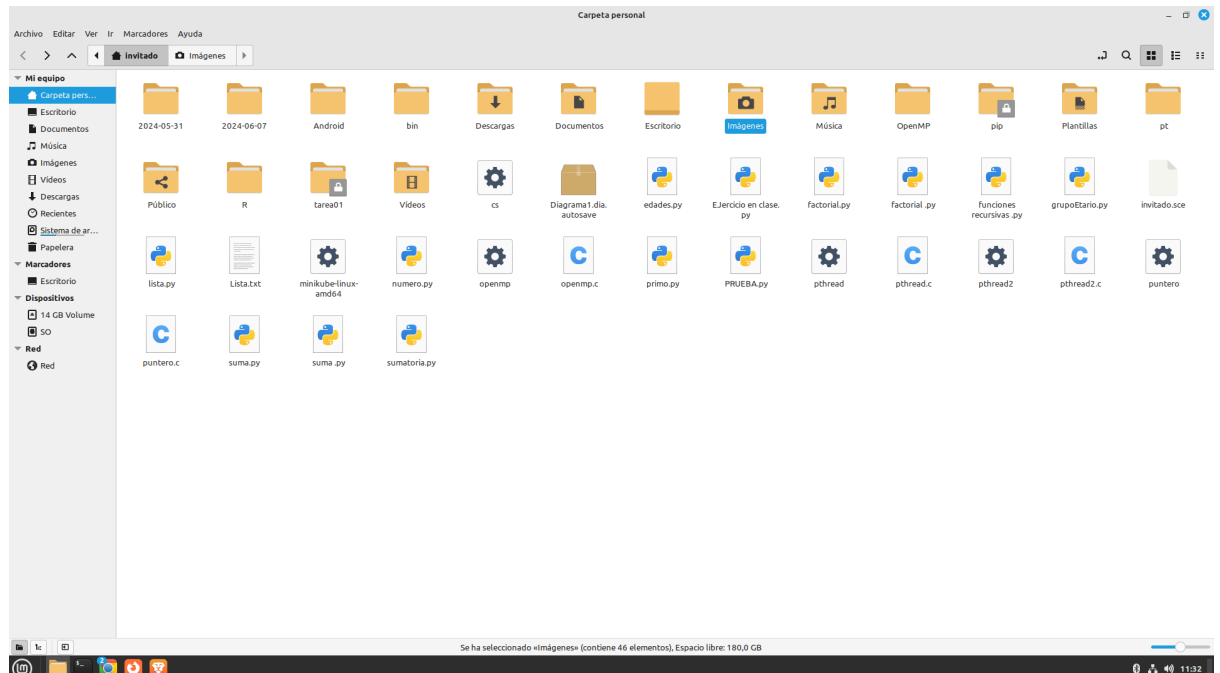
```
docker run -v ${PWD}:/data/ japeto/pujgcc:latest ls /data/
```

```
invitado@s2pc11:~$ docker run -v ${PWD}:/data/ japeto/puigcc:latest ls /data/
2024-05-31
2024-06-07
Android
Descargas
Diagrama1.dia.autosave
Documentos
Ejercicio en clase.py
Escritorio
Imágenes
Lista.txt
Música
OpenMP
PRUEBA.py
Plantillas
Público
R
Videos
bin
cs
edades.py
factorial .py
factorial.py
funciones recursivas .py
grupoEtario.py
invitado.sce
lista.py
minikube-linux-amd64
numero.py
openmp
openmp.c
pip
primo.py
pt
pthread
pthread.c
pthread2
pthread2.c
puntero
puntero.c
suma .py
suma.py
sumatoria.py
tarea01
invitado@s2pc11:~$
```

Question:

1. Is the displayed list the same as what is in your current folder?

Answer/ yes, it is



Finally, we can run C on a C or C++ file located in the Data folder. Change the name of the file to any of the provided files.

Paste a screenshot of result

```
docker run -v ${PWD}:/data/ japeto/pujgcc:latest gcc /data/helloworld.c / our  
version (docker run -v ${PWD}:/data/ japeto/pujgcc:latest g++ /data/lab.cpp)
```

```

invidado@s6pc28:~$ docker pull japeto/pujgcc:latest
latest: Pulling from japeto/pujgcc
aa18ad1a0d33: Pull complete
15a33158a136: Pull complete
f67323742a64: Pull complete
c4b45e832c38: Pull complete
e5d4afe2cf59: Pull complete
1efbd2d5674a: Pull complete
022a58c161f7: Pull complete
6461d3be7619: Pull complete
8287aab02fcd: Pull complete
edd3d1d6fad6: Pull complete
c6dbbe32db7e: Pull complete
Digest: sha256:9b3d7bbf410396d0a26e6bcffbb54e4239736d5a23ad694b03d93c26f9fc6868
Status: Downloaded newer image for japeto/pujgcc:latest
docker.io/japeto/pujgcc:latest

What's Next?
  1. Sign in to your Docker account → docker login
  2. View a summary of image vulnerabilities and recommendations → docker scout quickview japeto/pujgcc:latest
invidado@s6pc28:~$ mkdir so01
invidado@s6pc28:~$ cd so01
invidado@s6pc28:~/so01$ touch lab1
invidado@s6pc28:~/so01$ touch lab1.cpp
invidado@s6pc28:~/so01$ ls
lab1  lab1.cpp
invidado@s6pc28:~/so01$ nano lab1.cpp
invidado@s6pc28:~/so01$ docker run -v ${PWD}:/data/ japeto/pujgcc:latest gcc /data/lab1.cpp

/tmp/ccD43A1N.o: In function `__static_initialization_and_destruction_0(int, int)':
lab1.cpp:(.text+0x2e): undefined reference to `std::ios_base::Init::Init()'
lab1.cpp:(.text+0x3d): undefined reference to `std::ios_base::Init::~Init()'
collect2: error: ld returned 1 exit status
invidado@s6pc28:~/so01$ docker run -v ${PWD}:/data/ japeto/pujgcc:latest g++ /data/lab1.cpp
invidado@s6pc28:~/so01$ █

```

Restart and detach a container

Learn how to re-use a container where you installed something

Use the start command to restart the container created in the last exercise

Paste a screenshot of result

```
docker start -ti mycontainer /bin/bash
```

Go back to the container using the exec command instead of the run command.

Paste a screenshot of result

```
docker exec -ti mycontainer /bin/bash
```

Question :

1. What happens now when you exit the container? Is it stopped?

Answer

Check with and **Paste a screenshot of result**

```
docker ps -l
```

```
invitado@s6pc28:~$ docker ps -l
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS          PORTS          NAMES
7406d27f0f66   japeto/pujgcc  "bash --help"          4 minutes ago  Exited (0) 4 minutes ago          great_rubin
invitado@s6pc28:~$
```

In fact, the container keeps on running. This is because re-starting a container turns it into a “detached process” running in the background. Alternatively, we could have added the -d option to the first docker run command, creating directly a detached container.

Finally, you can stop the container.

```
docker stop mycontainer
```

This is the end of the practical session. We hope you enjoyed it. Don’t hesitate to ask any questions and feel free to contact us any time after the session!

List of commands

Search the available versions of an image in the Docker registry:

```
docker search
```

Pulling an image:

```
docker pull
```

Starting a container on a given image running a single command:

```
docker run -ti
```

Starting a container on a given image running a single command (detached):

```
docker run -d
```

List all containers and their status

```
docker ps -l
```

List all pulled images

```
docker images
```

Removing one local container

```
docker rm
```

Removing one local image

```
docker rmi
```

Clean all containers

```
docker rm $(docker ps -aq)
```

Clean all images (after cleaning the containers)

```
docker rmi $(docker images -aq)
```

Observations

- Deliveries must be made in teams of 4. Using a public repository on github and a pdf report
- If you do not understand the instructions for any of the activities, do not hesitate to jefferson.amado.pena@correounivalle.edu.co