CS 314 Operating Systems Lab

Lab Assignment 3

Hrishikesh Ravindra Karande   210010020

Part 1:

Q1] Modify the Minix3 source code such that the string “PID swapped in” is printed, whenever a user-level process is brought in by the scheduler.

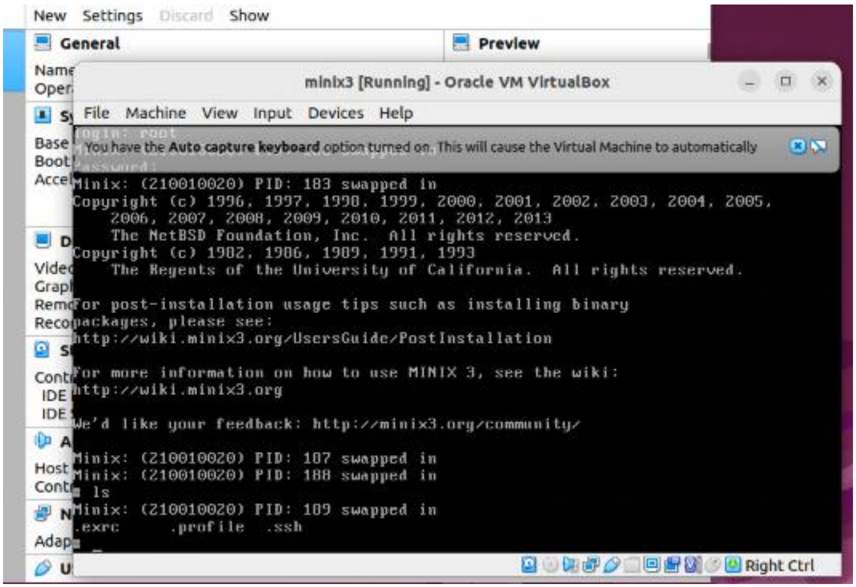
A] We need to modify the MINIX source code inside the directory /usr/src/minix/servers/sched in this directory the main.c file calls the do_start_scheduling(), do_stop_scheduling(), do_nice() and do_noquantum() functions which is located in the schedule.c file in same directory. Whenever a user level process in brought in by the scheduler the do_start_scheduling() function is called which in turn calls the schedule_process() function in the same file. This function is globally defined with the return type as int. In the schedule_process() function we set the values for the different fields of the schedproc object which is rmp.

We make the following modification to the file schedule.c in order to print whenever a new process is swapped in:

```
if (rmp->priority >= USER_Q) :

        printf("Minix: (210010020) PID: %d swapped in\n", _ENDPOINT_P(rmp->endpoint));
```

The if statement checks if the priority of the process is greater than equal to the USER_Q(which is defined in minix/include/minix/config.h) .

In order to make the changes we need to run the run.sh file given along with this using command **bash run.sh**. After running the script the following output can be seen here.


New
Settings
Discard
Show
General
Preview
minix3 [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
You have the **Auto capture keyboard** option turned on. This will cause the Virtual Machine to automatically
Right Ctrl

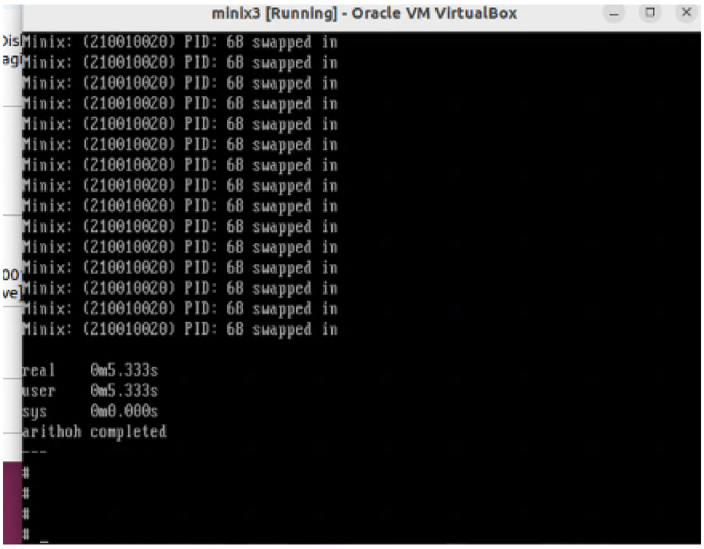
```
login: root
Password:
Minix: (210010020) PID: 183 swapped in
Copyright (c) 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005,
    2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013
    The NetBSD Foundation, Inc.  All rights reserved.
Copyright (c) 1982, 1986, 1989, 1991, 1993
    The Regents of the University of California.  All rights reserved.

For post-installation usage tips such as installing binary
packages, please see:
http://wiki.minix3.org/UsersGuide/PostInstallation

For more information on how to use MINIX 3, see the wiki:
http://wiki.minix3.org

We'd like your feedback: http://minix3.org/community/

Minix: (210010020) PID: 187 swapped in
Minix: (210010020) PID: 188 swapped in
# ls
Minix: (210010020) PID: 189 swapped in
.exrc      .profile   .ssh
# _
```

Part2

To classify the processes as I/o Bound or CPU bound we need to examine the user and sys times of the processes.

- CPU-Bound Process:
  - A process is considered CPU-bound if it spends a significant portion of its time using the CPU.
  - If the user time is significantly higher than the sys time, it suggests that the process is consuming a lot of CPU time.
- I/O-Bound Process:
  - A process is considered I/O-bound if it spends a significant portion of its time waiting for I/O operations (e.g., disk I/O, network I/O).
  - If the sys time is relatively higher than the user time, it might indicate that the process is spending more time in system mode, possibly waiting for I/O operations to complete.

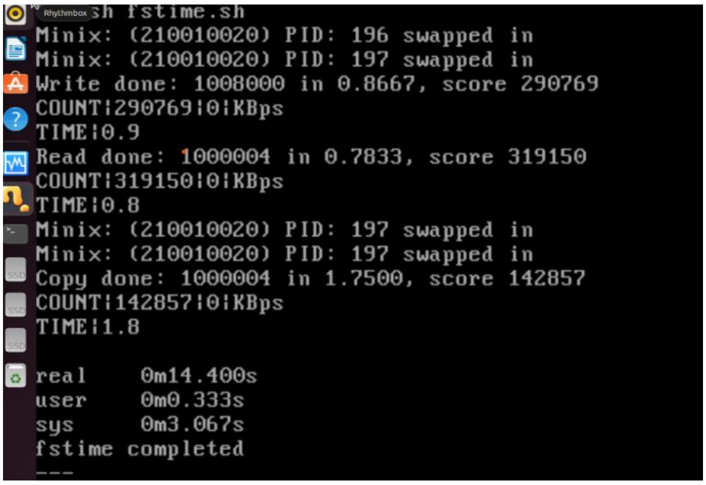
1. arithoh.sh:



```
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in

real    0m5.333s
user    0m5.333s
sys     0m0.000s
arithoh completed
---
#
#
#
# _
```

- The process with pid 68 runs when arithoh.sh script is executed.
- The process spends 5.33s in user mode and 0.00s in kernel mode, total wall clock time spent = 5.33s.
- This is CPU Bound Process.

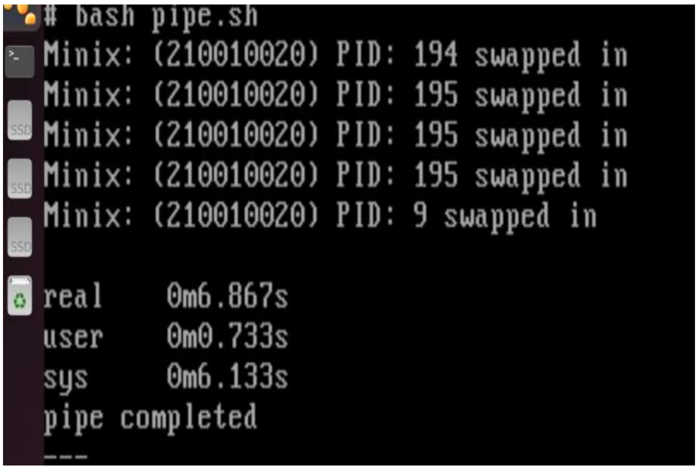
2. fstime.sh



```
sh fstime.sh
Minix: (210010020) PID: 196 swapped in
Minix: (210010020) PID: 197 swapped in
Write done: 1008000 in 0.8667, score 290769
COUNT|290769|0|KBps
TIME|0.9
Read done: 1000004 in 0.7833, score 319150
COUNT|319150|0|KBps
TIME|0.8
Minix: (210010020) PID: 197 swapped in
Minix: (210010020) PID: 197 swapped in
Copy done: 1000004 in 1.7500, score 142857
COUNT|142857|0|KBps
TIME|1.8

real    0m14.400s
user    0m0.333s
sys     0m3.067s
fstime completed
---
```

- The process with pid 196 runs when fstime.sh script is executed.
- The process spends 0.333s in user mode and 3.067s in kernel mode, total wall clock time spent = 14.400s.
- This is I/O Bound Process as it waits for the Input for long time.

3. pipe.sh



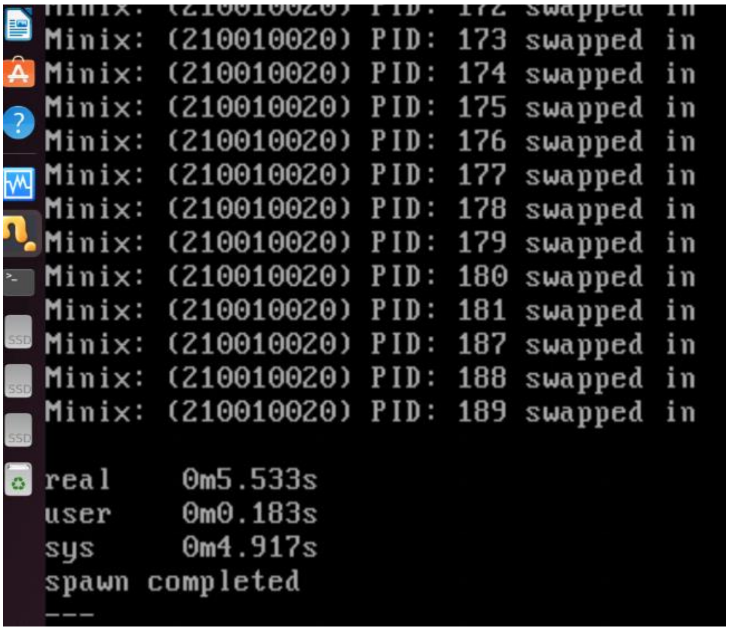
```
# bash pipe.sh
Minix: (210010020) PID: 194 swapped in
Minix: (210010020) PID: 195 swapped in
Minix: (210010020) PID: 195 swapped in
Minix: (210010020) PID: 195 swapped in
Minix: (210010020) PID: 9 swapped in

real    0m6.867s
user    0m0.733s
sys     0m6.133s
pipe completed
---
```

- The process with pid 194,195 and 9 runs when pipe.sh script is executed.
- The process spends 0.733s in user mode and 6.133s in kernel mode, total wall clock time spent = 6.867.

From the code, it appears to involve both CPU-bound and user-bound aspects.

The CPU is actively engaged in the loop iterations, and the handling of interrupts indicates potential user-bound characteristics due to interactions with the operating system.

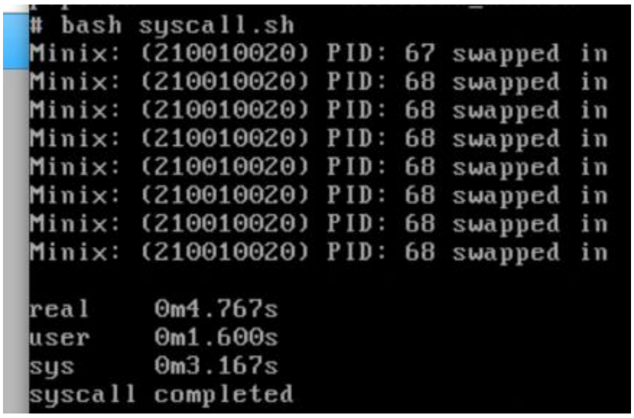
4. **spawn.sh:**



```
Minix: (210010020) PID: 172 swapped in
Minix: (210010020) PID: 173 swapped in
Minix: (210010020) PID: 174 swapped in
Minix: (210010020) PID: 175 swapped in
Minix: (210010020) PID: 176 swapped in
Minix: (210010020) PID: 177 swapped in
Minix: (210010020) PID: 178 swapped in
Minix: (210010020) PID: 179 swapped in
Minix: (210010020) PID: 180 swapped in
Minix: (210010020) PID: 181 swapped in
Minix: (210010020) PID: 187 swapped in
Minix: (210010020) PID: 188 swapped in
Minix: (210010020) PID: 189 swapped in

real     0m5.533s
user     0m0.183s
sys      0m4.917s
spawn completed
---
```

- The process with pid 172,173,174,175,176,177,178,179,180,181,187,188 ,189 runs when spawn.sh script is executed.
- The process spends 0.733s in user mode and 6.133s in kernel mode, total wall clock time spent = 6.867.
- Based on the analysis, the code is more likely to be CPU-bound. The main workload involves creating and terminating processes in a loop, which is a CPU-intensive operation. The absence of explicit I/O operations in the loop supports the classification as CPU-bound.

5. **syscall.sh**

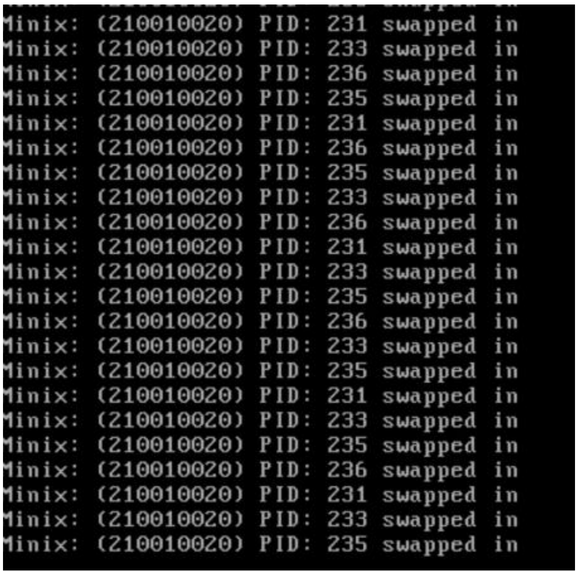


```
# bash syscall.sh
Minix: (210010020) PID: 67 swapped in
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in
Minix: (210010020) PID: 68 swapped in

real     0m4.767s
user     0m1.600s
sys      0m3.167s
syscall completed
```

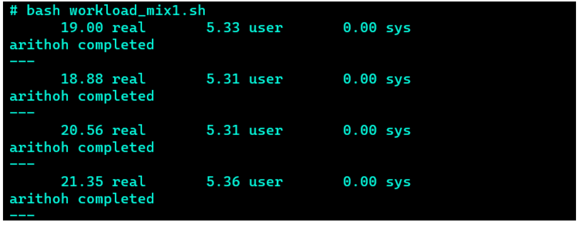
- The process with pid 67 and 68 runs when syscall.sh script is executed.
- The process spends 1.600s in user mode and 3.167s in kernel mode, total wall clock time spent = 4.767s.

This is The code exhibits a mix of CPU-bound and user-bound characteristics, depending on the specific calls:"mix" includes a mix of CPU-bound and user-bound operations whereas close" and "getpid" are more user-bound due to the nature of the system calls. "exec" involves process creation and execution, which includes both CPU-bound and user-bound aspects.

6. workbench_mix1.sh



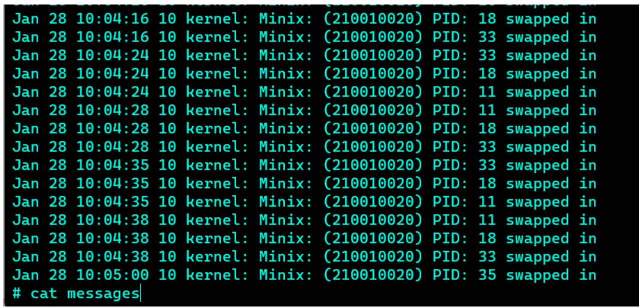
```
Minix: (210010020) PID: 231 swapped in
Minix: (210010020) PID: 233 swapped in
Minix: (210010020) PID: 236 swapped in
Minix: (210010020) PID: 235 swapped in
Minix: (210010020) PID: 231 swapped in
Minix: (210010020) PID: 236 swapped in
Minix: (210010020) PID: 235 swapped in
Minix: (210010020) PID: 233 swapped in
Minix: (210010020) PID: 236 swapped in
Minix: (210010020) PID: 231 swapped in
Minix: (210010020) PID: 233 swapped in
Minix: (210010020) PID: 235 swapped in
Minix: (210010020) PID: 236 swapped in
Minix: (210010020) PID: 233 swapped in
Minix: (210010020) PID: 235 swapped in
Minix: (210010020) PID: 231 swapped in
Minix: (210010020) PID: 233 swapped in
Minix: (210010020) PID: 235 swapped in
Minix: (210010020) PID: 236 swapped in
Minix: (210010020) PID: 231 swapped in
Minix: (210010020) PID: 233 swapped in
Minix: (210010020) PID: 235 swapped in
```



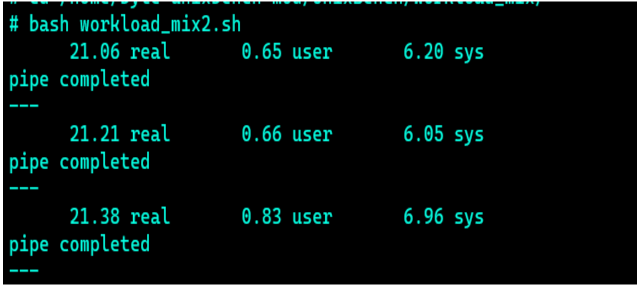
```
# bash workload_mix1.sh
       19.00 real        5.33 user        0.00 sys
arithoh completed
---
       18.88 real        5.31 user        0.00 sys
arithoh completed
---
       20.56 real        5.31 user        0.00 sys
arithoh completed
---
       21.35 real        5.36 user        0.00 sys
arithoh completed
---
```

- As seen above arithoh.sh is CPU Bound process.
- Four instances of arithoh.sh run initiated run one after the other until their time slice is over.
- Processes 231,233,236 and 235 correspond to 4 instances of arithoh.sh.

7. workload_mix2.sh



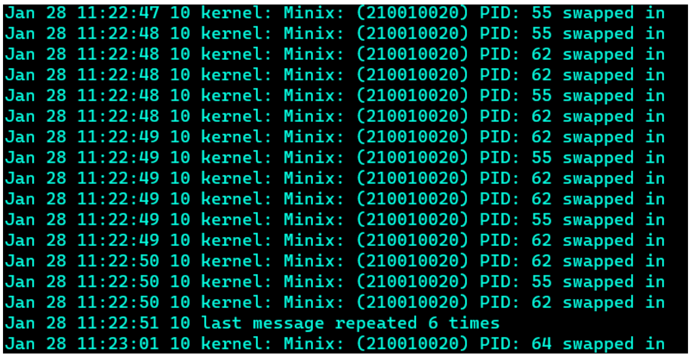
```
Jan 28 10:04:16 10 kernel: Minix: (210010020) PID: 18 swapped in
Jan 28 10:04:16 10 kernel: Minix: (210010020) PID: 33 swapped in
Jan 28 10:04:24 10 kernel: Minix: (210010020) PID: 33 swapped in
Jan 28 10:04:24 10 kernel: Minix: (210010020) PID: 18 swapped in
Jan 28 10:04:24 10 kernel: Minix: (210010020) PID: 11 swapped in
Jan 28 10:04:28 10 kernel: Minix: (210010020) PID: 11 swapped in
Jan 28 10:04:28 10 kernel: Minix: (210010020) PID: 18 swapped in
Jan 28 10:04:28 10 kernel: Minix: (210010020) PID: 33 swapped in
Jan 28 10:04:35 10 kernel: Minix: (210010020) PID: 33 swapped in
Jan 28 10:04:35 10 kernel: Minix: (210010020) PID: 18 swapped in
Jan 28 10:04:35 10 kernel: Minix: (210010020) PID: 11 swapped in
Jan 28 10:04:38 10 kernel: Minix: (210010020) PID: 11 swapped in
Jan 28 10:04:38 10 kernel: Minix: (210010020) PID: 18 swapped in
Jan 28 10:04:38 10 kernel: Minix: (210010020) PID: 33 swapped in
Jan 28 10:05:00 10 kernel: Minix: (210010020) PID: 35 swapped in
# cat messages
```



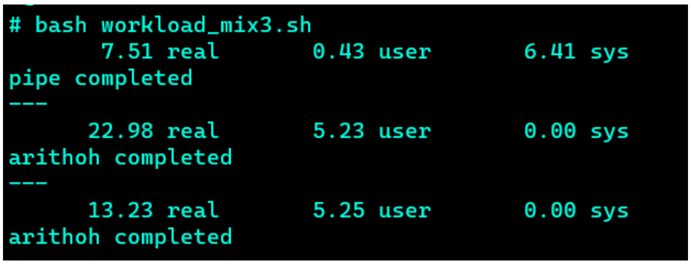
```
# bash workload_mix2.sh
       21.06 real        0.65 user        6.20 sys
pipe completed
---
       21.21 real        0.66 user        6.05 sys
pipe completed
---
       21.38 real        0.83 user        6.96 sys
pipe completed
---
```

- Each instance of pipe.sh is a I/O bound process as explained earlier.
- When executed process waits for the input meanwhile others who have the input use CPU in round robin manner.

8. workload_mix3.sh



```
Jan 28 11:22:47 10 kernel: Minix: (210010020) PID: 55 swapped in
Jan 28 11:22:48 10 kernel: Minix: (210010020) PID: 55 swapped in
Jan 28 11:22:48 10 kernel: Minix: (210010020) PID: 62 swapped in
Jan 28 11:22:48 10 kernel: Minix: (210010020) PID: 62 swapped in
Jan 28 11:22:48 10 kernel: Minix: (210010020) PID: 55 swapped in
Jan 28 11:22:48 10 kernel: Minix: (210010020) PID: 62 swapped in
Jan 28 11:22:49 10 kernel: Minix: (210010020) PID: 62 swapped in
Jan 28 11:22:49 10 kernel: Minix: (210010020) PID: 55 swapped in
Jan 28 11:22:49 10 kernel: Minix: (210010020) PID: 62 swapped in
Jan 28 11:22:49 10 kernel: Minix: (210010020) PID: 62 swapped in
Jan 28 11:22:49 10 kernel: Minix: (210010020) PID: 55 swapped in
Jan 28 11:22:49 10 kernel: Minix: (210010020) PID: 62 swapped in
Jan 28 11:22:50 10 kernel: Minix: (210010020) PID: 62 swapped in
Jan 28 11:22:50 10 kernel: Minix: (210010020) PID: 55 swapped in
Jan 28 11:22:50 10 kernel: Minix: (210010020) PID: 62 swapped in
Jan 28 11:22:51 10 last message repeated 6 times
Jan 28 11:23:01 10 kernel: Minix: (210010020) PID: 64 swapped in
```



```
# bash workload_mix3.sh
        7.51 real        0.43 user        6.41 sys
pipe completed
---
       22.98 real        5.23 user        0.00 sys
arithoh completed
---
       13.23 real        5.25 user        0.00 sys
arithoh completed
```

- When arithoh.h and pipe are initiated, one is CPU bound and other is I/O bound.
- While the I/O bound waits for an input from the user the cpu bound process uses the CPU.
- After pipe completes its execution it leaves.