1) Create and compile your own program in C (named as PA01) and run strace

```
#include <stdio.h>

int main(void) {
	FILE *fd;
	float pval=3.14159;
	int i,k;

	If ((fd= fopen("myTstFile","r+"))==NULL)
		printf("\n Program Failed, figure out why...\n");
	else{
		printf("\n Simple pie value %1.8f\n", pval);
		for (i=0; i<100; i++) {
			k = rand()%10;
			if (fprintf(fd, "%f\n",pval+i*k)==-1) perror("write err"); fflush(fd);
			printf("."); fflush(stdout);
			}
		fclose(fd); printf("\n Program successful ends\n");
		}
	}
```

The following steps were taken to answer this question:

1. strace ./PA01 *(output included in the attachments of this document)*
2. strace –o PA01acalls.txt -c ./PA01

Below are the top 5 most frequent system calls and a brief explanation of what they do:

UNIQUE	COUNT	Explanation of system call
mmap	8	This is a POSIX (IEEE standard) Compliant Unix System Call that maps files or devices into memory. It establishes a mapping between a process's address space and a file or shared memory object.[1]q
pread64	4	This is formerly known as pread(), reads from a file descriptor at a given position (or offset) without changing the file pointer. Hence the p prefix before read()[2]
brk	3	This is a UNIX memory management system call to control the amount of memory allocated to the data segment of the process. They specifically are used to change dynamically the amount of space allocated for the calling process's data segment. [3]
openat	3	This is identical to open() except that the path argument is interpreted relative tot eh starting point implied by the flides argument. A specific value can be put into the flides argument (AT_FDCWD) to resolve the path relative to the current working directory. If the flides is ignored than the path is absolute. The open() establishes a connection between a file and a file descriptor. The file descriptor is used by other I/O functions to refer to the file. [4]
newfstatat	3	newfstatat() obtains file attributes similar to the stat(), lstat(), and fstat() functions. It has the flides argument like openat() where the directory may be relative to the current working directory or absolute. It also has an argument to behave like lstat() where it does not automatically follow symbolic links. The base behavior of this function [similar to stat(), lstat(), and fstat()] is to obtain information about the file. Such as directories, symbolic links, and file descriptors. Ultimately it gets the file status. [5]

Among the five system calls, openat caused the program to fail as myTstFile does not exist. Further evidence is shown on line 36 in the file PA01a.txt, which shows we receive an ENOENT error message and a return value of –1. This ENOENT error means a certain file should exist but does not and is inaccessible.

3. The table below was created using process 1(a) with the exception that myTstFile was created and in the same directory as PA01 compiled C file. *(Output included in the attachments of this document)*

   Note the question request the most frequently invoked system call, albeit, the top 5 were provided again for consistency.

UNIQUE	COUNT	DESCRIPTION
write	204	The write() function attempts to write nbyte bytes from the buffer point to by buf to the file associated with the open file descriptor, flides. If nbyte is 0, write() will return 0 and have no other results if the file is a regular file. [6]
mmap	8	See 1(a)
newfstatat	4	See 1(a)
pread64	4	See 1(a)
brk	3	See 1(a)

fopen was determined to be a library call based on the table in:
man man

```
The table below shows the section numbers of the manual followed by the types of pages they contain.

1   Executable programs or shell commands
2   System calls (functions provided by the kernel)
3   Library calls (functions within program libraries)
4   Special files (usually found in /dev)
5   File formats and conventions, e.g. /etc/passwd
6   Games
7   Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7), man-pages(7)
8   System administration commands (usually only for root)
9   Kernel routines [Non standard]
```

man fopen yielded the following results and correlates to the open(2) – system call.

```
FOPEN(3)                                          Linux Programmer's Manual

NAME
       fopen, fdopen, freopen - stream open functions

SEE ALSO
       open(2), fclose(3), fileno(3), fmemopen(3), fopencookie(3), open_memstream(3)
```

Referencing the same *man man* table above.

printf was determined to be an library call within the manual.

```
SEE ALSO
       printf(3)

       Full documentation <https://www.gnu.org/software/coreutils/printf>
       or available locally via: info '(coreutils) printf invocation'
```

printf(3) is associated with a number of other library calls as is shown below:

SEE ALSO top

printf(1), asprintf(3), puts(3), scanf(3), setlocale(3), strfromd(3), wcrtomb(3), wprintf(3), locale(5)

These library calls often call the write(2) system call ultimately.

2) STRACE a Linux utility command -- cal. Run "strace –c cal", capture the output, and then pick the top three system calls which consumed the system time and briefly describe their functionality.

strace -o Question2.text -c cal

% time	seconds	usecs/ call	calls	errors	syscall	Description
27.33	0.000434	**31**	14	0	mmap	This is a POSIX (IEEE standard) Compliant Unix System Call that maps files or devices into memory. It establishes a mapping between a process's address space and a file or shared memory object. [1]
16.06	0.000255	42	6	0	openat	This is identical to open() except that the path argument is interpreted relative tot eh starting point implied by the flides argument. A specific value can be put into the flides argument (AT_FDCWD) to resolve the path relative to the current working directory. If the flides is ignored than the path is absolute. The open() establishes a between a file and a file descriptor. The file descriptor is used by other I/O functions to refer to the file. [4]
12.85	0.000204	12	9	0	newfstatat	newfstatat() obtains file attributes similar to the stat(), lstat(), and fstat() functions. It has the flides argument like openat() where the directory may be relative to the current working directory or absolute. It also has an argument to behave like lstat() where it does not automatically follow symbolic links. The base behavior of this function [similar to stat(), lstat(), and fstat()] is to obtain information about the file. Such as directories, symbolic links, and file descriptors. Ultimately it gets the file status. [5]

3) STRACE/LTRACE Linux utility commands "ls". Command ltrace is another tracing tool used for tracing the library function calls. Use both strace and ltrace to Linux command ls to report what library functions and system calls are used to

The following commands were issued within Ubuntu 20.04 LTS and did not yield successful results.

- *ltrace -o Question3-ltrace.txt ls*
- *ltrace -o ls*
- *ltrace –c ls*
- *uftrace ls*
- *uftrace -p ls*

These commands did not work because *ltrace* and *uftrace* both require the library calls or system calls to be compiled in a specific way and the 20.04 LTS version of Ubuntu does compiles the source code for these in a non-compatible way. For example: see the required gcc compiling flags for *ls* to work with *uftrace*. See screen shots below:

```
⊗ amming Assignments - PA)/PA01/Code$ uftrace ls
  uftrace: ./cmds/record.c:1625:check_binary
   ERROR: Can't find 'mcount' symbol in the '/usr/bin/ls'.
          It seems not to be compiled with -pg or -finstrument-functions flag.
          You can rebuild your program with it or use -P option for dynamic tracing.
```

```
% time     seconds  usecs/call     calls      function
------ ----------- ----------- --------- --------------------
------ ----------- ----------- --------- --------------------
100.00    0.000000                     0 total
```

The following command functioned as intended and was used to answer the subsequent questions.
strace -o Question3-strace.txt ls

a) Open the current directory

Referencing the output from strace ls, line 6.

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

The openat() function is identical to the open() function except that the path argument is interpreted relative to the starting point implied by the fildes argument. If the fildes argument has the special value AT_FDCWD, a relative path argument will be resolved relative to the current working directory.[4]

b) Get the list of directory entries

getdents64(3, 0x55998c5539f0 /* 15 entries */, 32768) = 504
getdents64(3, 0x55998c5539f0 /* 0 entries */, 32768) = 0

The getdents() and getdents64() functions - get directory entries. The original Linux getdents() system call did not handle large filesystems and large file offsets. Consequently, Linux 2.4 added getdents64(), with wider types for the d_ino and d_off fields. In addition, getdents64() supports an explicit d_type field. The getdents64() system call is like getdents(), except that its second argument is a pointer to a buffer.[10]

c) Print the output to your screen

write(1, " CompiledPA01 PA01\t PA01.docx\t"..., 79) = 79
write(1, " myTstFile PA01.c\t'PA01 Not"..., 101) = 101

I wrote to a file in my use of trace but without the *-o filename.txt* argument it will print to the terminal.
As is shown above this is write(1)

The write(1) function sends a message to another user.
[6], [11]

4) Bibliography

[1] "mmap - man pages section 2: System Calls." https://docs.oracle.com/cd/E36784_01/html/E36872/mmap-2.html#scrolltoc (accessed Sep. 05, 2022).
[2] "pread - man pages section 2: System Calls." https://docs.oracle.com/cd/E36784_01/html/E36872/pread-2.html#scrolltoc (accessed Sep. 05, 2022).
[3] "brk - man pages section 2: System Calls." https://docs.oracle.com/cd/E36784_01/html/E36872/brk-2.html#scrolltoc (accessed Sep. 05, 2022).
[4] "openat - man pages section 2: System Calls." https://docs.oracle.com/cd/E36784_01/html/E36872/openat-2.html (accessed Sep. 05, 2022).
[5] "fstatat - man pages section 2: System Calls." https://docs.oracle.com/cd/E36784_01/html/E36872/fstatat-2.html#scrolltoc (accessed Sep. 05, 2022).
[6] "write - man pages section 2: System Calls." https://docs.oracle.com/cd/E36784_01/html/E36872/write-2.html#scrolltoc (accessed Sep. 05, 2022).
[7] "fopen(3) - Linux manual page." https://man7.org/linux/man-pages/man3/freopen.3.html (accessed Sep. 05, 2022).
[8] "open(2) - Linux manual page." https://man7.org/linux/man-pages/man2/open.2.html (accessed Sep. 05, 2022).
[9] "CS170 Lecture notes -- What do you C?" https://sites.cs.ucsb.edu/~rich/class/cs170/notes/SystemCalls/index.html (accessed Sep. 05, 2022).
[10] "getdents(2) - Linux manual page." https://man7.org/linux/man-pages/man2/getdents.2.html (accessed Sep. 06, 2022).
[11] "write(1) - Linux manual page." https://man7.org/linux/man-pages/man1/write.1.html (accessed Sep. 06, 2022).