ECE 8400 / ENGI 9875 Lab 4: paging

Mohammed Shakeel, Tousif Habib

TOTAL POINTS

22 / 22

QUESTION 1

Prelab 5 pts

1.1 Major vs minor page faults 1 / 1

✓ **- 0 pts** *Correct*

1.2 get_rusage_string() 2 / 2

✓ **- 0 pts** *Correct*

1.3 balloon() 2 / 2

✓ **- 0 pts** *Correct*

- 1 pts Incomplete in filling up the allocated memory

- 1 pts Missing the test output and explanation

QUESTION 2

CLI 4 pts

2.1 Execute get_rusage_string() 2 / 2

✓ **- 0 pts** *Correct*

- 1 pts Missing the expected output

- 1 pts Missing the evidence of the work/code/explanation

2.2 Plot page faults 2 / 2

✓ **- 0 pts** *Correct*

- 1 pts Missing the plot for major page fault

- 2 pts Not answered

QUESTION 3

GUI 8 pts

3.1 Execute and explain differences 2 / 2

✓ **- 0 pts** *Correct*

- 1 pts Missing the evidence of the work and output

- 2 pts Not answered (or pages not linked)

3.2 get_command_rusage() 2 / 2

✓ **- 0 pts** *Correct*

- 2 pts Not submitted/Failed to complete the section

- 1 pts Missing the evidence of the work

- 1 pts Missing the explanation of the observations

- 2 pts Not answered (or pages not linked)

3.3 on_click_run_*() 4 / 4

✓ **- 0 pts** *Correct*

- 4 pts Not submitted / Failed to complete the section

- 2 pts Missing the plots

- 1 pts Missing the explanation

- 0.5 pts Not enough evaluation

- 1 pts Incomplete explanation

- 1 pts Missing evidence of the work and outputs

- 4 pts Not answered (or pages not linked)

QUESTION 4

Memory pressure 5 pts

4.1 Memory availability 1 / 1

✓ **- 0 pts** *Correct*

4.2 Run under gdb 2 / 2

✓ **- 0 pts** *Correct*

- 1 pts Missing plots showing the page faults

4.3 Balloon out memory 1 / 1

✓ **- 0 pts** *Correct*

- 1 pts Incomplete section/Failed to finish the section

- 1 pts Not answered (or pages not linked)

4.4 Run under gdb again 1 / 1

✓ **- 0 pts** *Correct*

- 0.5 pts Missing the clarity in showing the major and minor page faults in the plot

- 0.5 pts Missing the representation of page faults

- 1 pts Missing the evidence of the work and outputs

- 1 pts Not answered (or pages not linked)

Lab 4

Prelab

1. According to the getrusage(2) manual page, a major page fault is one that can only be satisfied by accessing the disk. A minor page fault, on the other hand, can be satisfied by sharing pages that are already in memory.

   A minor fault means the page is in memory but not allocated to the requesting process or not marked as present in the memory management unit. A major fault means the page is no longer in memory.

2.

```c
#include <stdio.h>
#include <sys/resource.h>
#include <sys/time.h>

void get_rusage_string(char *buffer, size_t buffer_len) {
    struct rusage usage;
    getrusage(RUSAGE_SELF, &usage);
    snprintf(buffer, buffer_len,
             "User time: %ld.%06ld\nSystem time: %ld.%06ld\nMax resident
set size: %ld\n",
             usage.ru_utime.tv_sec, usage.ru_utime.tv_usec,
             usage.ru_stime.tv_sec, usage.ru_stime.tv_usec,
             usage.ru_maxrss);
}
```

1.1 Major vs minor page faults **1 / 1**

✓ **- 0 pts** *Correct*

Lab 4

Prelab

1. According to the getrusage(2) manual page, a major page fault is one that can only be satisfied by accessing the disk. A minor page fault, on the other hand, can be satisfied by sharing pages that are already in memory.

   A minor fault means the page is in memory but not allocated to the requesting process or not marked as present in the memory management unit. A major fault means the page is no longer in memory.

2.

```c
#include <stdio.h>
#include <sys/resource.h>
#include <sys/time.h>

void get_rusage_string(char *buffer, size_t buffer_len) {
    struct rusage usage;
    getrusage(RUSAGE_SELF, &usage);
    snprintf(buffer, buffer_len,
             "User time: %ld.%06ld\nSystem time: %ld.%06ld\nMax resident
set size: %ld\n",
             usage.ru_utime.tv_sec, usage.ru_utime.tv_usec,
             usage.ru_stime.tv_sec, usage.ru_stime.tv_usec,
             usage.ru_maxrss);
}
```

1.2 get_rusage_string() **2 / 2**

✓ **- 0 pts** *Correct*

3.

```c
#include <err.h>
#include <stdbool.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/mman.h>
#include <sys/resource.h>

// Utility function to get current memory usage
static long get_memory_usage() {
    struct rusage usage;
    if (getrusage(RUSAGE_SELF, &usage) != 0) {
        err(EXIT_FAILURE, "getrusage failed");
    }
    return usage.ru_maxrss;
}

static void balloon(size_t bytes) {
    char *data = malloc(bytes);
    if (data == NULL) {
        errx(EXIT_FAILURE, "Failed to allocate %zu bytes", bytes);
    }

    for (size_t i = 0; i < bytes; i += sysconf(_SC_PAGESIZE)) {
        data[i] = 1;
    }

    if (mlock(data, bytes) != 0) {
        err(EXIT_FAILURE, "mlock failed");
    }
}

int main(int argc, char *argv[]) {
    size_t Gigs;

    while (true) {
        printf("Enter number of GiB to fill: ");

        char *line;
        size_t n = 0;
        if (getline(&line, &n, stdin) < 0) {
            err(-1, "failed to read number from stdin");
        }
```

```
        char *end;
        Gigs = strtol(line, &end, 10);

        free(line);

        if (end != line) {
            break;
        }

        fprintf(stderr, "Invalid number: '%s'\n", line);
    }

    long before_memory_usage = get_memory_usage();
    printf("Memory usage before ballooning: %ld KiB\n", before_memory_usage);

    for (size_t i = 0; i < Gigs; i++) {
        balloon(1024 * 1024 * 1024);
    }

    long after_memory_usage = get_memory_usage();
    printf("Memory usage after ballooning: %ld KiB\n", after_memory_usage);

    printf("Ballooning complete. Press Enter to quit.");
    getchar();

    return 0;
}
```

Output:

```
Tousifs-MacBook-Pro:Lab4 tousifhabib$ gcc -o balloon balloon.c
Tousifs-MacBook-Pro:Lab4 tousifhabib$ ./balloon
Enter number of GiB to fill: 2
Memory usage before ballooning: 1097728 KiB
Memory usage after ballooning: 2148597760 KiB
```

In this example, we can see that the memory usage increased after the ballooning process, indicating that the balloon() function indeed used the requested number of bytes of physical memory.

The balloon() function works as follows:

1. Allocate memory of the requested size using malloc().
2. Touch each memory page to ensure it is backed by physical memory (using sysconf(_SC_PAGESIZE) to get the system's page size).
3. Lock the memory using mlock() to prevent it from being paged out to disk (swap).

1.3 balloon() **2 / 2**

✓ **- 0 pts** *Correct*

- 1 pts Incomplete in filling up the allocated memory

- 1 pts Missing the test output and explanation

Procedure

Setup

. Ensuring the following exists:
Gcc compiler:

1. **GCC C++ Compiler**

```
msshakeel@slbncsf2112pc03:~$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/9/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none:hsa
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu
9.4.0-1ubuntu1~20.04.1'
--with-bugurl=file:///usr/share/doc/gcc-9/README.Bugs
--enable-languages=c,ada,c++,go,brig,d,fortran,objc,obj-c++,gm2
--prefix=/usr --with-gcc-major-version-only --program-suffix=-9
--program-prefix=x86_64-linux-gnu- --enable-shared --enable-linker-build-id
--libexecdir=/usr/lib --without-included-gettext --enable-threads=posix
--libdir=/usr/lib --enable-nls --enable-clocale=gnu
--enable-libstdcxx-debug --enable-libstdcxx-time=yes
--with-default-libstdcxx-abi=new --enable-gnu-unique-object
--disable-vtable-verify --enable-plugin --enable-default-pie
--with-system-zlib --with-target-system-zlib=auto --enable-objc-gc=auto
--enable-multiarch --disable-werror --with-arch-32=i686 --with-abi=m64
--with-multilib-list=m32,m64,mx32 --enable-multilib --with-tune=generic
--enable-offload-targets=nvptx-none=/build/gcc-9-Av3uEd/gcc-9-9.4.0/debian/
tmp-nvptx/usr,hsa --without-cuda-driver --enable-checking=release
--build=x86_64-linux-gnu --host=x86_64-linux-gnu --target=x86_64-linux-gnu
Thread model: posix
gcc version 9.4.0 (Ubuntu 9.4.0-1ubuntu1~20.04.1)
```

Libgtk

```
msshakeel@slbncsf2112pc03:~$ dpkg -l libgtk* | grep -e '^i' | grep -e
'libgtk-*[0-9]'
ii  libgtk-3-0:amd64         3.24.20-0ubuntu1.1 amd64     GTK graphical
user interface library
ii  libgtk-3-bin             3.24.20-0ubuntu1.1 amd64     programs for the
GTK graphical user interface library
ii  libgtk-3-common          3.24.20-0ubuntu1.1 all       common files for
the GTK graphical user interface library
ii  libgtk2-perl             2:1.24993-1ubuntu2 amd64     Perl interface
to the 2.x series of the Gimp Toolkit library
ii  libgtk2.0-0:amd64        2.24.32-4ubuntu4   amd64     GTK graphical
user interface library - old version
ii  libgtk2.0-0:i386         2.24.32-4ubuntu4   i386      GTK graphical
user interface library - old version
ii  libgtk2.0-bin            2.24.32-4ubuntu4   amd64     programs for the
GTK graphical user interface library
ii  libgtk2.0-common         2.24.32-4ubuntu4   all       common files for
the GTK graphical user interface library
ii  libgtk2.0-dev:amd64      2.24.32-4ubuntu4   amd64     development
files for the GTK library
ii  libgtk3-perl             0.037-1          all         Perl bindings
for the GTK+ graphical user interface library
```

Cmake

```
msshakeel@slbncsf2112pc03:~$ cmake -version
cmake version 3.16.3

CMake suite maintained and supported by Kitware (kitware.com/cmake).
```

Ninja:

```
msshakeel@slbncsf2112pc03:~$ ninja --version
1.10.0
```

3.

```
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4$ ./setup-build.sh
+ mkdir -p build
+ cd build
+ cmake -G Ninja ..
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Failed
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
-- Found PkgConfig: /usr/bin/pkg-config (found version "0.29.1")
-- Checking for module 'gtk+-2.0'
--   Found gtk+-2.0, version 2.24.32
-- Configuring done
-- Generating done
-- Build files have been written to:
/users/labnet/st3/msshakeel/operatingLabs/lab4/build
+ echo OK, the build is ready. cd build and run ninja next.
OK, the build is ready. cd build and run ninja next.
```

4.

```
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4/build$ ninja
[7/8] Building C object gui/CMakeFiles/gui.dir/gtk-ui.c.o
In file included from /usr/include/gtk-2.0/gtk/gtkobject.h:37,
                 from /usr/include/gtk-2.0/gtk/gtkwidget.h:36,
                 from /usr/include/gtk-2.0/gtk/gtkcontainer.h:35,
                 from /usr/include/gtk-2.0/gtk/gtkbin.h:35,
                 from /usr/include/gtk-2.0/gtk/gtkwindow.h:36,
                 from /usr/include/gtk-2.0/gtk/gtkdialog.h:35,
                 from /usr/include/gtk-2.0/gtk/gtkaboutdialog.h:32,
                 from /usr/include/gtk-2.0/gtk/gtk.h:33,
                 from ../gui/gtk-ui.c:21:
/usr/include/gtk-2.0/gtk/gtktypeutils.h:236:1: warning: 'GTypeDebugFlags'
is deprecated [-Wdeprecated-declarations]
  236 | void            gtk_type_init   (GTypeDebugFlags    debug_flags);
      | ^~~~
In file included from /usr/include/glib-2.0/gobject/gobject.h:24,
                 from /usr/include/glib-2.0/gobject/gbinding.h:29,
                 from /usr/include/glib-2.0/glib-object.h:22,
                 from /usr/include/glib-2.0/gio/gioenums.h:28,
                 from /usr/include/glib-2.0/gio/giotypes.h:28,
                 from /usr/include/glib-2.0/gio/gio.h:26,
                 from /usr/include/gtk-2.0/gdk/gdkapplaunchcontext.h:30,
                 from /usr/include/gtk-2.0/gdk/gdk.h:32,
                 from /usr/include/gtk-2.0/gtk/gtk.h:32,
                 from ../gui/gtk-ui.c:21:
/usr/include/glib-2.0/gobject/gtype.h:679:1: note: declared here
  679 | {
      | ^
In file included from /usr/include/gtk-2.0/gtk/gtktoolitem.h:31,
                 from /usr/include/gtk-2.0/gtk/gtktoolbutton.h:30,
                 from /usr/include/gtk-2.0/gtk/gtkmenutoolbutton.h:30,
                 from /usr/include/gtk-2.0/gtk/gtk.h:126,
                 from ../gui/gtk-ui.c:21:
/usr/include/gtk-2.0/gtk/gtktooltips.h:73:3: warning: 'GTimeVal' is
deprecated: Use 'GDateTime' instead [-Wdeprecated-declarations]
   73 |   GTimeVal last_popdown;
      |   ^~~~~~~~
In file included from /usr/include/glib-2.0/glib/galloca.h:32,
                 from /usr/include/glib-2.0/glib.h:30,
                 from /usr/include/glib-2.0/gobject/gbinding.h:28,
                 from /usr/include/glib-2.0/glib-object.h:22,
                 from /usr/include/glib-2.0/gio/gioenums.h:28,
```

```
                 from /usr/include/glib-2.0/gio/giotypes.h:28,
                 from /usr/include/glib-2.0/gio/gio.h:26,
                 from /usr/include/gtk-2.0/gdk/gdkapplaunchcontext.h:30,
                 from /usr/include/gtk-2.0/gdk/gdk.h:32,
                 from /usr/include/gtk-2.0/gtk/gtk.h:32,
                 from ../gui/gtk-ui.c:21:
/usr/include/glib-2.0/glib/gtypes.h:547:8: note: declared here
  547 | struct _GTimeVal
      |     ^~~~~~~~
[8/8] Linking C shared library gui/libgui.so
```

5.

```
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4$ dd if=/dev/urandom bs=1024
count=10240 | base64 > random.txt
10240+0 records in
10240+0 records out
10485760 bytes (10 MB, 10 MiB) copied, 0.104484 s, 100 MB/s
```

Sample from random.txt:

```
gi/tTovWrrI7kvweFhIUxBCdwHAh45Syt8YC6nSE03lTKE1PgW8hbjhPn4+i3PigccJonQvOB
+WrnVMETJJ20AaOw4ntW5+iNSUaz+Z24h6frfdoLcEl4v0aO+DLkTiVXp2dJXSSIKEwdaHDcYFQ
P
kEzz7jclyKhxVE+pJOfvySWl2Qj/YnO6YGrsfteAnEaIlxZ1mpI/mNBayvTbKazmjotBokcW5zE
3
PQXBJizuyhOrjT6gqJRlht+a5Bc2gez6majXleWqh5YGP/sCfO8puUfXaLGpwjbuBkGEoEIm7nG
2
f/OGY760n1Kg29WuVFr384oJdDq6jppa8YTc0gG1kZuC65VU+1SQ6L1bA2mbT4YPROtQHbDcrEE
n
3sD7CK6izvdN0l1Z+0XRWnAuOOyoE8zmiW/22w+AguMztKDrSz8L6kC5b6JEXxc27ZTz7Phu8dy
m
gMizQTkHFVMgnxD3xhqvnrqz8p1aowd0zPyWbj/3ubNwpw44k7OsDWFQfmG8RgnsOaNhdSQqTaf
a
aWiVncEa/i58Cj2LhlldC41MRDPYh1Br6RCn635WlzMoDc3N0Dk5bykR8VPPoqdGW8keZTcI9/+
f
rb8+aP4fMXKzVQlOYgZQDKSDOH6oOEDKBZb19ZLkxCqvPs6JQolP/d1kyX334E1ETDdfntofgO1
o
mnSHQy5qdyL9+XjwI4LW25dDvzO0oJD4f5BKSHV3pM6O3ecjAYT6BrWOzNx6I92FbHnjxVB/f60
```

```
C
mkDq0itJ7UmhJxyQuhDrXLJVhWZAlM/n7eSqkaemOVb3mZ+CHLi6UXcY29em1lZVJfeHiVlDgwx
N
cBrPagUwEUnkOLch3MUOpNlYbBU0d1yHH1Crk76VuhbcO11aZrqh+Kz5mylBq4lyeofvIJRgSxz
b
tIvJtgSvUNTurYDL0o+vNyi9hYeK5smSulAOyXb8VcYabYiChGplv0FtIj+wUI6XqA
```

Command Line lab tool

1.

```
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4/build$ ./lab
----
beginning of main()
----
rusage:
User time: 0.001820
      System time: 0.000000
 Max resident set size: 1532
 Max page faults: 0
 Min page faults: 92

----
after ui_init()
----
rusage:
User time: 0.001948
      System time: 0.000000
 Max resident set size: 1532
 Max page faults: 0
 Min page faults: 99

Press Enter to exit...
----------------------
after starting UI thread
----------------------
rusage information
User time: 0.002108
      System time: 0.000000
 Max resident set size: 1532
 Max page faults: 0
```

2.1 Execute get_rusage_string() **2 / 2**

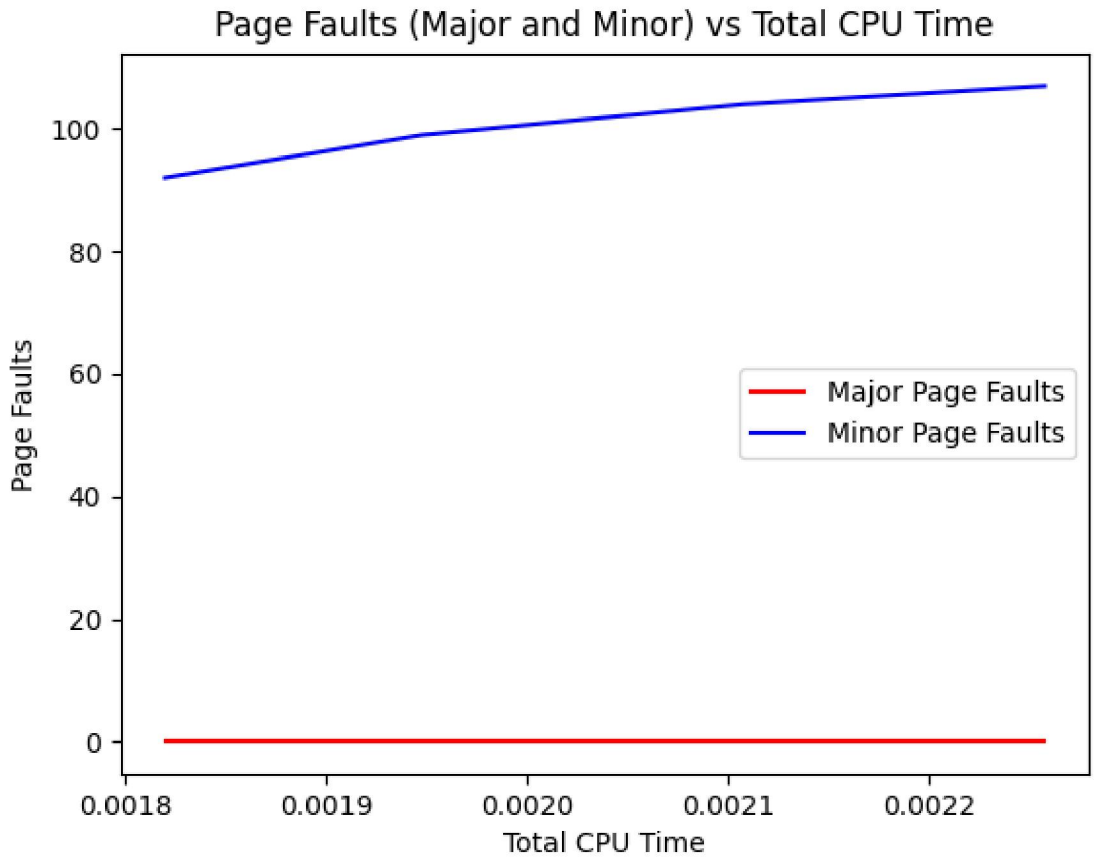
✓ **- 0 pts** *Correct*

- 1 pts Missing the expected output

- 1 pts Missing the evidence of the work/code/explanation

```
 Min page faults: 104



----
after UI thread complete
----
rusage:
User time: 0.002258
      System time: 0.000000
 Max resident set size: 1532
 Max page faults: 0
 Min page faults: 107
```

2.



2.2 Plot page faults **2 / 2**

✓ **- 0 pts** *Correct*

- 1 pts Missing the plot for major page fault

- 2 pts Not answered

3.

```
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4$ ./setup-build.sh
+ mkdir -p build
+ cd build
+ cmake -G Ninja ..
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Failed
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
-- Found PkgConfig: /usr/bin/pkg-config (found version "0.29.1")
-- Checking for module 'gtk+-2.0'
--   Found gtk+-2.0, version 2.24.32
-- Configuring done
-- Generating done
-- Build files have been written to:
/users/labnet/st3/msshakeel/operatingLabs/lab4/build
+ echo OK, the build is ready. cd build and run ninja next.
OK, the build is ready. cd build and run ninja next.
```

GUI

1.

```
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4/build$
LD_PRELOAD=gui/libgui.so ./lab
----
beginning of main()
----
rusage:
User time: 0.000000
      System time: 0.010450
 Max resident set size: 6368
 Max page faults: 0
 Min page faults: 479

----
after ui_init()
----
rusage:
User time: 0.007694
      System time: 0.023084
 Max resident set size: 15028
 Max page faults: 0
 Min page faults: 939

----
after UI thread complete
----
rusage:
User time: 0.051623
      System time: 0.035739
 Max resident set size: 24804
 Max page faults: 15
 Min page faults: 1659
```

The differences between the outputs of the previous and the current tool invocations are:

- User time and system time values have increased in the current invocation, indicating that the program took more time to execute with the LD_PRELOAD environment variable set to gui/libgui.so.

- Max resident set size values are significantly higher in the current invocation, suggesting that the program used more memory when LD_PRELOAD was set to gui/libgui.so.

- The number of minor page faults has increased in the current invocation, indicating more memory pages needed to be loaded from disk to memory.

- The number of major page faults has changed from zero in the previous invocation to 15 in the current invocation after the UI thread is complete. This means that the memory pages were not readily available and had to be fetched from the swap area, causing the program to slow down.

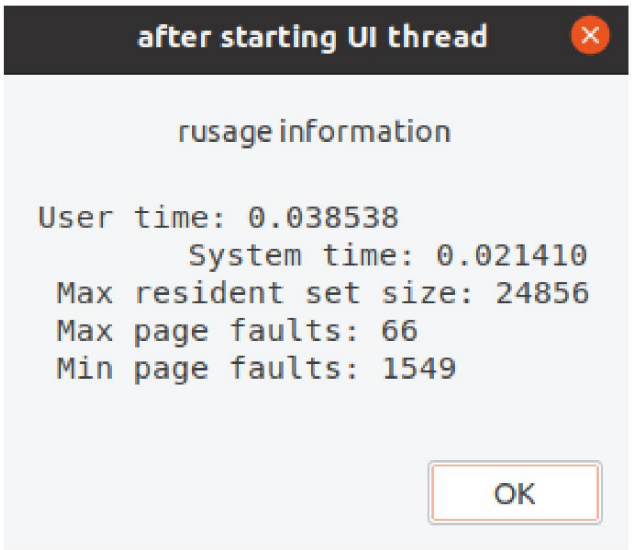
Opening and cancelling the exit dialog would result in increased resource usage. In this case, the rusage values show an increase in user time, system time, max resident set size, and the number of major and minor page faults. This suggests that more CPU time and memory resources were used when the exit dialog was opened and then cancelled. The exit dialog requires additional resources to load and render the graphical interface, and manage the user interaction events. Cancelling the exit dialog might also involve additional resource management and cleanup, which contributes to the increased resource usage.

2.

```
void get_command_rusage(int argc, char * const argv[], char *buffer, size_t
len)
{
   assert(argv[argc] == NULL);
pid_t pid = fork();
   if (pid == 0) {
       // child process
       execvp(argv[0], argv);
       _exit(1);
   } else if (pid > 0) {
       // parent process
       int status;
       struct rusage usage;
       wait4(pid, &status, 0, &usage);
       // format usage data into buffer
       snprintf(buffer, len, "user time: %ld.%06lds\n"
                             "system time: %ld.%06lds\n"
```

```
                            "max resident set size: %ld\n"
                            "integral shared memory size: %ld\n"
                            "integral unshared data size: %ld\n"
                            "integral unshared stack size: %ld\n"
                            "page reclaims (soft page faults): %ld\n"
                            "page faults (hard page faults): %ld\n"
                            "swaps: %ld\n"
                            "block input operations: %ld\n"
                            "block output operations: %ld\n",
                usage.ru_utime.tv_sec, usage.ru_utime.tv_usec,
                usage.ru_stime.tv_sec, usage.ru_stime.tv_usec,
                usage.ru_maxrss,
                usage.ru_ixrss,
                usage.ru_idrss,
                usage.ru_isrss,
                usage.ru_minflt,
                usage.ru_majflt,
                usage.ru_nswap,
                usage.ru_inblock,
                usage.ru_oublock);
    }

}
```

Testing the program:



Output:

```
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4/build$
LD_PRELOAD=gui/libgui.so ./lab
```

```
----
beginning of main()
----
rusage:
User time: 0.005067
      System time: 0.002533
 Max resident set size: 6316
 Max page faults: 5
 Min page faults: 477

----
after ui_init()
----
rusage:
User time: 0.016292
      System time: 0.012219
 Max resident set size: 15484
 Max page faults: 48
 Min page faults: 910



----
after UI thread complete
----
rusage:
User time: 0.167330
      System time: 0.043182
 Max resident set size: 25736
 Max page faults: 73
 Min page faults: 1640msshakeel@slbncsf2112pc03:~/operatingLabs/lab4/build$
LD_PRELOAD=gui/libgui.so ./lab
----
beginning of main()
----
rusage:
User time: 0.005067
      System time: 0.002533
 Max resident set size: 6316
 Max page faults: 5
 Min page faults: 477

----
```

```
after ui_init()
----
rusage:
User time: 0.016292
      System time: 0.012219
 Max resident set size: 15484
 Max page faults: 48
 Min page faults: 910



----
after UI thread complete
----
rusage:
User time: 0.167330
      System time: 0.043182
 Max resident set size: 25736
 Max page faults: 73
 Min page faults: 1640
```

The implementation of get_command_rusage() is designed to execute a given command, specified by argc and argv, and record the resource usage of that command using wait4(2). The function first forks a new process, and the child process runs the specified command using execvp(). The parent process waits for the child process to complete using wait4() and retrieves the resource usage information via the rusage struct.

In the console output, the following can be observed:

- At the beginning of main(), the user time is 0.005067 seconds, the system time is 0.002533 seconds, max resident set size is 6316, major page faults are 5, and minor page faults are 477.

- After ui_init(), the user time has increased to 0.016292 seconds, system time has increased to 0.012219 seconds, max resident set size has increased to 15484, major page faults have increased to 48, and minor page faults have increased to 910.

- After the UI thread is complete, the user time further increased to 0.167330 seconds, system time to 0.043182 seconds, max resident set size to 25736, major page faults to 73, and minor page faults to 1640.

Comparing the results from the previous invocation, we can see that the user time, system time, max resident set size, major page faults, and minor page faults have all increased in this case as well. This is expected since the program is now executing with the LD_PRELOAD=gui/libgui.so environment variable set, which affects the resource usage.

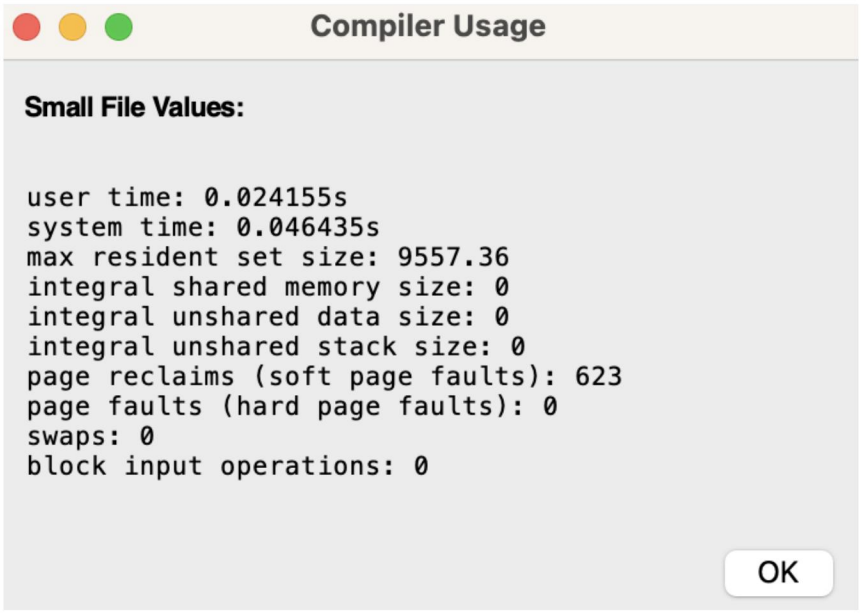
Hence, the implementation of get_command_rusage() effectively measures the resource usage of the given command by utilizing the wait4(2) function. The console output demonstrates an increase in resource usage when executing the program with LD_PRELOAD=gui/libgui.so, highlighting the impact of the environment variable on the program's performance.

3. Fixed on_click_run_grep_large and on_click_run_grep_small functions:

```
static void
on_click_run_grep_large(GtkWidget *_widget, gpointer _data)
{
    // Create a large file using dd command
    system("dd if=/dev/random of=large_file bs=1M count=100");

    char * const args[] = {
        "grep",
        "pattern", // Replace 'pattern' with the desired pattern to search
for
        "large_file",
        NULL
    };

    char buffer[4096];
    get_command_rusage(3, args, buffer, sizeof(buffer));

    dialog_show("Grep Large File Usage", "Executed grep on large_file",
buffer);
}

static void
on_click_run_grep_small(GtkWidget *widget, gpointer data)
{
    // Create a small file using dd command
    system("dd if=/dev/random of=small_file bs=1K count=100");

    char * const args[] = {
        "grep",
        "pattern", // Replace 'pattern' with the desired pattern to search
```

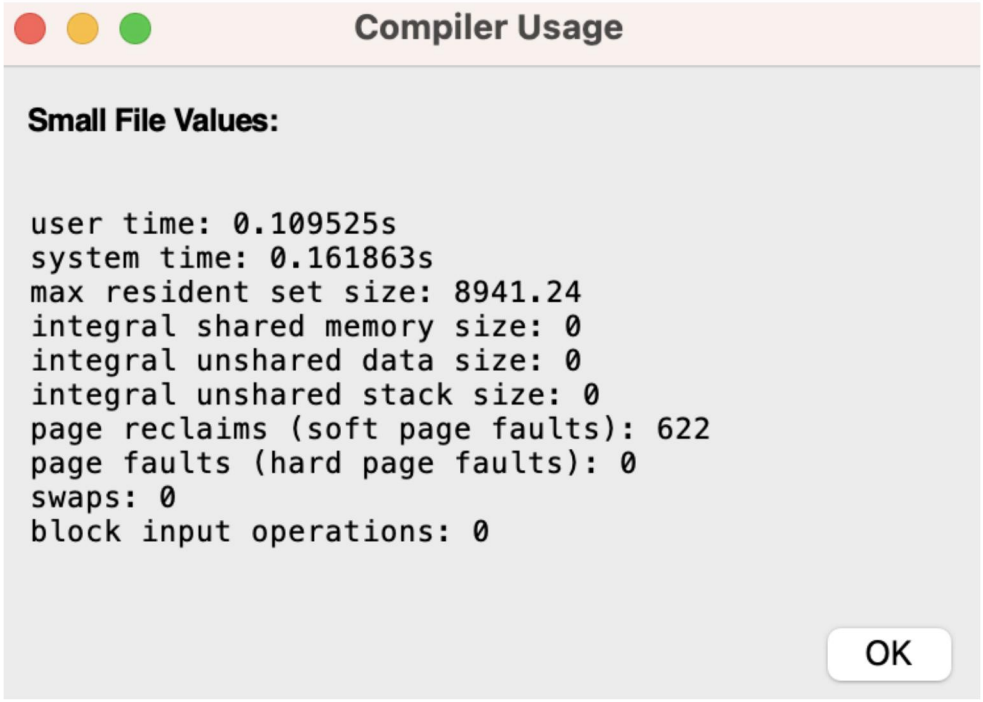
```
for
        "small_file",
        NULL
    };

    char buffer[4096];
    get_command_rusage(3, args, buffer, sizeof(buffer));

    dialog_show("Grep Small File Usage", "Executed grep on small_file",
buffer);
}
```

Values for small files:

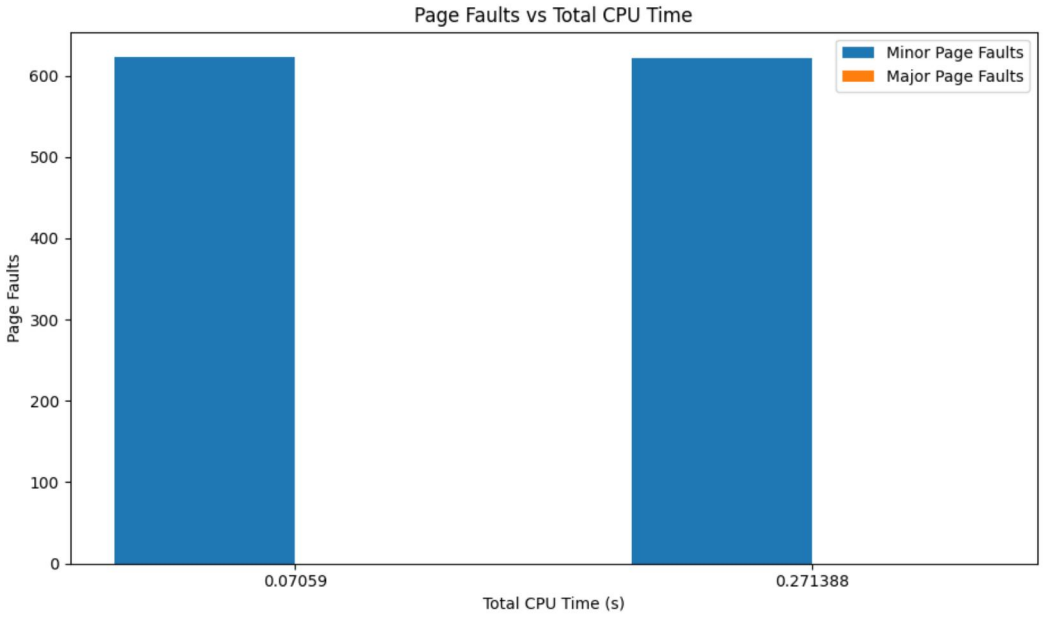



```
user time: 0.024155s
system time: 0.046435s
max resident set size: 9557.36
integral shared memory size: 0
integral unshared data size: 0
integral unshared stack size: 0
page reclaims (soft page faults): 623
page faults (hard page faults): 0
swaps: 0
block input operations: 0
block output operations: 0
```

Values for large files:




```
user time: 0.109525s
system time: 0.161863s
max resident set size: 8941.24
integral shared memory size: 0
integral unshared data size: 0
integral unshared stack size: 0
page reclaims (soft page faults): 622
page faults (hard page faults): 0
swaps: 0
block input operations: 0
block output operations: 0
```



Observations and differences:

- User time: The user time is higher for the large file compared to the small file, as grep needs more time to process the larger file.
- System time: Similarly, the system time is higher for the large file, as it requires more file system interaction to read the content.
- Max resident set size: The max resident set size for the small file is slightly higher than the large file. This difference could be due to variations in memory management and allocation during the execution of grep on different sized files. However, the difference is not significant and does not indicate a strong correlation between file size and memory consumption for grep.
- Page reclaims (soft page faults): The values are quite similar, with only a difference of 1, which indicates that the memory allocation and deallocation patterns are not significantly affected by the file size.

Overall, the main differences between running grep against small and large files lie in the user and system times, as expected due to the increased time required to process and interact with larger files. The memory consumption and page faults are not significantly impacted by the file size in this case.

3.1 Execute and explain differences **2 / 2**

✓ **- 0 pts** *Correct*

- 1 pts Missing the evidence of the work and output

- 2 pts Not answered (or pages not linked)

3.

```
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4$ ./setup-build.sh
+ mkdir -p build
+ cd build
+ cmake -G Ninja ..
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Failed
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
-- Found PkgConfig: /usr/bin/pkg-config (found version "0.29.1")
-- Checking for module 'gtk+-2.0'
--   Found gtk+-2.0, version 2.24.32
-- Configuring done
-- Generating done
-- Build files have been written to:
/users/labnet/st3/msshakeel/operatingLabs/lab4/build
+ echo OK, the build is ready. cd build and run ninja next.
OK, the build is ready. cd build and run ninja next.
```

4.

```
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4/build$ ninja
[7/8] Building C object gui/CMakeFiles/gui.dir/gtk-ui.c.o
In file included from /usr/include/gtk-2.0/gtk/gtkobject.h:37,
                 from /usr/include/gtk-2.0/gtk/gtkwidget.h:36,
                 from /usr/include/gtk-2.0/gtk/gtkcontainer.h:35,
                 from /usr/include/gtk-2.0/gtk/gtkbin.h:35,
                 from /usr/include/gtk-2.0/gtk/gtkwindow.h:36,
                 from /usr/include/gtk-2.0/gtk/gtkdialog.h:35,
                 from /usr/include/gtk-2.0/gtk/gtkaboutdialog.h:32,
                 from /usr/include/gtk-2.0/gtk/gtk.h:33,
                 from ../gui/gtk-ui.c:21:
/usr/include/gtk-2.0/gtk/gtktypeutils.h:236:1: warning: 'GTypeDebugFlags'
is deprecated [-Wdeprecated-declarations]
  236 | void            gtk_type_init   (GTypeDebugFlags    debug_flags);
      | ^~~~
In file included from /usr/include/glib-2.0/gobject/gobject.h:24,
                 from /usr/include/glib-2.0/gobject/gbinding.h:29,
                 from /usr/include/glib-2.0/glib-object.h:22,
                 from /usr/include/glib-2.0/gio/gioenums.h:28,
                 from /usr/include/glib-2.0/gio/giotypes.h:28,
                 from /usr/include/glib-2.0/gio/gio.h:26,
                 from /usr/include/gtk-2.0/gdk/gdkapplaunchcontext.h:30,
                 from /usr/include/gtk-2.0/gdk/gdk.h:32,
                 from /usr/include/gtk-2.0/gtk/gtk.h:32,
                 from ../gui/gtk-ui.c:21:
/usr/include/glib-2.0/gobject/gtype.h:679:1: note: declared here
  679 | {
      | ^
In file included from /usr/include/gtk-2.0/gtk/gtktoolitem.h:31,
                 from /usr/include/gtk-2.0/gtk/gtktoolbutton.h:30,
                 from /usr/include/gtk-2.0/gtk/gtkmenutoolbutton.h:30,
                 from /usr/include/gtk-2.0/gtk/gtk.h:126,
                 from ../gui/gtk-ui.c:21:
/usr/include/gtk-2.0/gtk/gtktooltips.h:73:3: warning: 'GTimeVal' is
deprecated: Use 'GDateTime' instead [-Wdeprecated-declarations]
   73 |   GTimeVal last_popdown;
      |   ^~~~~~~~
In file included from /usr/include/glib-2.0/glib/galloca.h:32,
                 from /usr/include/glib-2.0/glib.h:30,
                 from /usr/include/glib-2.0/gobject/gbinding.h:28,
                 from /usr/include/glib-2.0/glib-object.h:22,
                 from /usr/include/glib-2.0/gio/gioenums.h:28,
```

The differences between the outputs of the previous and the current tool invocations are:

- User time and system time values have increased in the current invocation, indicating that the program took more time to execute with the LD_PRELOAD environment variable set to gui/libgui.so.

- Max resident set size values are significantly higher in the current invocation, suggesting that the program used more memory when LD_PRELOAD was set to gui/libgui.so.

- The number of minor page faults has increased in the current invocation, indicating more memory pages needed to be loaded from disk to memory.

- The number of major page faults has changed from zero in the previous invocation to 15 in the current invocation after the UI thread is complete. This means that the memory pages were not readily available and had to be fetched from the swap area, causing the program to slow down.

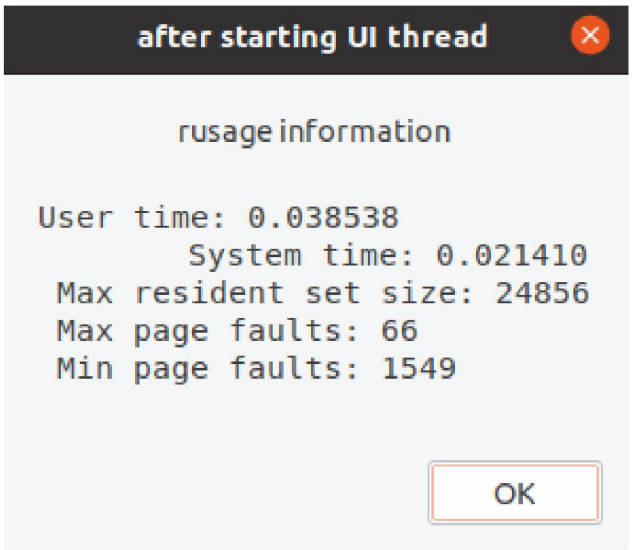
Opening and cancelling the exit dialog would result in increased resource usage. In this case, the rusage values show an increase in user time, system time, max resident set size, and the number of major and minor page faults. This suggests that more CPU time and memory resources were used when the exit dialog was opened and then cancelled. The exit dialog requires additional resources to load and render the graphical interface, and manage the user interaction events. Cancelling the exit dialog might also involve additional resource management and cleanup, which contributes to the increased resource usage.

2.

```
void get_command_rusage(int argc, char * const argv[], char *buffer, size_t
len)
{
   assert(argv[argc] == NULL);
pid_t pid = fork();
   if (pid == 0) {
       // child process
       execvp(argv[0], argv);
       _exit(1);
   } else if (pid > 0) {
       // parent process
       int status;
       struct rusage usage;
       wait4(pid, &status, 0, &usage);
       // format usage data into buffer
       snprintf(buffer, len, "user time: %ld.%06lds\n"
                             "system time: %ld.%06lds\n"
```

```
                            "max resident set size: %ld\n"
                            "integral shared memory size: %ld\n"
                            "integral unshared data size: %ld\n"
                            "integral unshared stack size: %ld\n"
                            "page reclaims (soft page faults): %ld\n"
                            "page faults (hard page faults): %ld\n"
                            "swaps: %ld\n"
                            "block input operations: %ld\n"
                            "block output operations: %ld\n",
                usage.ru_utime.tv_sec, usage.ru_utime.tv_usec,
                usage.ru_stime.tv_sec, usage.ru_stime.tv_usec,
                usage.ru_maxrss,
                usage.ru_ixrss,
                usage.ru_idrss,
                usage.ru_isrss,
                usage.ru_minflt,
                usage.ru_majflt,
                usage.ru_nswap,
                usage.ru_inblock,
                usage.ru_oublock);
    }

}
```

Testing the program:



Output:

```
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4/build$
LD_PRELOAD=gui/libgui.so ./lab
```

```
----
beginning of main()
----
rusage:
User time: 0.005067
      System time: 0.002533
 Max resident set size: 6316
 Max page faults: 5
 Min page faults: 477

----
after ui_init()
----
rusage:
User time: 0.016292
      System time: 0.012219
 Max resident set size: 15484
 Max page faults: 48
 Min page faults: 910



----
after UI thread complete
----
rusage:
User time: 0.167330
      System time: 0.043182
 Max resident set size: 25736
 Max page faults: 73
 Min page faults: 1640msshakeel@slbncsf2112pc03:~/operatingLabs/lab4/build$
LD_PRELOAD=gui/libgui.so ./lab
----
beginning of main()
----
rusage:
User time: 0.005067
      System time: 0.002533
 Max resident set size: 6316
 Max page faults: 5
 Min page faults: 477

----
```

```
after ui_init()
----
rusage:
User time: 0.016292
      System time: 0.012219
 Max resident set size: 15484
 Max page faults: 48
 Min page faults: 910



----
after UI thread complete
----
rusage:
User time: 0.167330
      System time: 0.043182
 Max resident set size: 25736
 Max page faults: 73
 Min page faults: 1640
```

The implementation of get_command_rusage() is designed to execute a given command, specified by argc and argv, and record the resource usage of that command using wait4(2). The function first forks a new process, and the child process runs the specified command using execvp(). The parent process waits for the child process to complete using wait4() and retrieves the resource usage information via the rusage struct.

In the console output, the following can be observed:

- At the beginning of main(), the user time is 0.005067 seconds, the system time is 0.002533 seconds, max resident set size is 6316, major page faults are 5, and minor page faults are 477.

- After ui_init(), the user time has increased to 0.016292 seconds, system time has increased to 0.012219 seconds, max resident set size has increased to 15484, major page faults have increased to 48, and minor page faults have increased to 910.

- After the UI thread is complete, the user time further increased to 0.167330 seconds, system time to 0.043182 seconds, max resident set size to 25736, major page faults to 73, and minor page faults to 1640.

3.2 get_command_rusage() **2 / 2**

✓ **- 0 pts** *Correct*

- 2 pts Not submitted/Failed to complete the section

- 1 pts Missing the evidence of the work

- 1 pts Missing the explanation of the observations

- 2 pts Not answered (or pages not linked)

3.

```
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4$ ./setup-build.sh
+ mkdir -p build
+ cd build
+ cmake -G Ninja ..
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Failed
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
-- Found PkgConfig: /usr/bin/pkg-config (found version "0.29.1")
-- Checking for module 'gtk+-2.0'
--   Found gtk+-2.0, version 2.24.32
-- Configuring done
-- Generating done
-- Build files have been written to:
/users/labnet/st3/msshakeel/operatingLabs/lab4/build
+ echo OK, the build is ready. cd build and run ninja next.
OK, the build is ready. cd build and run ninja next.
```

```
                 from /usr/include/glib-2.0/gio/giotypes.h:28,
                 from /usr/include/glib-2.0/gio/gio.h:26,
                 from /usr/include/gtk-2.0/gdk/gdkapplaunchcontext.h:30,
                 from /usr/include/gtk-2.0/gdk/gdk.h:32,
                 from /usr/include/gtk-2.0/gtk/gtk.h:32,
                 from ../gui/gtk-ui.c:21:
/usr/include/glib-2.0/glib/gtypes.h:547:8: note: declared here
  547 | struct _GTimeVal
      |     ^~~~~~~~
[8/8] Linking C shared library gui/libgui.so
```

5.

```
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4$ dd if=/dev/urandom bs=1024
count=10240 | base64 > random.txt
10240+0 records in
10240+0 records out
10485760 bytes (10 MB, 10 MiB) copied, 0.104484 s, 100 MB/s
```

Sample from random.txt:

```
gi/tTovWrrI7kvweFhIUxBCdwHAh45Syt8YC6nSE03lTKE1PgW8hbjhPn4+i3PigccJonQvOB
+WrnVMETJJ20AaOw4ntW5+iNSUaz+Z24h6frfdoLcEl4v0aO+DLkTiVXp2dJXSSIKEwdaHDcYFQ
P
kEzz7jclyKhxVE+pJOfvySWl2Qj/YnO6YGrsfteAnEaIlxZ1mpI/mNBayvTbKazmjotBokcW5zE
3
PQXBJizuyhOrjT6gqJRlht+a5Bc2gez6majXleWqh5YGP/sCfO8puUfXaLGpwjbuBkGEoEIm7nG
2
f/OGY760n1Kg29WuVFr384oJdDq6jppa8YTc0gG1kZuC65VU+1SQ6L1bA2mbT4YPROtQHbDcrEE
n
3sD7CK6izvdN0l1Z+0XRWnAuOOyoE8zmiW/22w+AguMztKDrSz8L6kC5b6JEXxc27ZTz7Phu8dy
m
gMizQTkHFVMgnxD3xhqvnrqz8p1aowd0zPyWbj/3ubNwpw44k7OsDWFQfmG8RgnsOaNhdSQqTaf
a
aWiVncEa/i58Cj2LhlldC41MRDPYh1Br6RCn635WlzMoDc3N0Dk5bykR8VPPoqdGW8keZTcI9/+
f
rb8+aP4fMXKzVQlOYgZQDKSDOH6oOEDKBZb19ZLkxCqvPs6JQolP/d1kyX334E1ETDdfntofgO1
o
mnSHQy5qdyL9+XjwI4LW25dDvzO0oJD4f5BKSHV3pM6O3ecjAYT6BrWOzNx6I92FbHnjxVB/f60
```

```
C
mkDq0itJ7UmhJxyQuhDrXLJVhWZAlM/n7eSqkaemOVb3mZ+CHLi6UXcY29em1lZVJfeHiVlDgwx
N
cBrPagUwEUnkOLch3MUOpNlYbBU0d1yHH1Crk76VuhbcO11aZrqh+Kz5mylBq4lyeofvIJRgSxz
b
tIvJtgSvUNTurYDL0o+vNyi9hYeK5smSulAOyXb8VcYabYiChGplv0FtIj+wUI6XqA
```

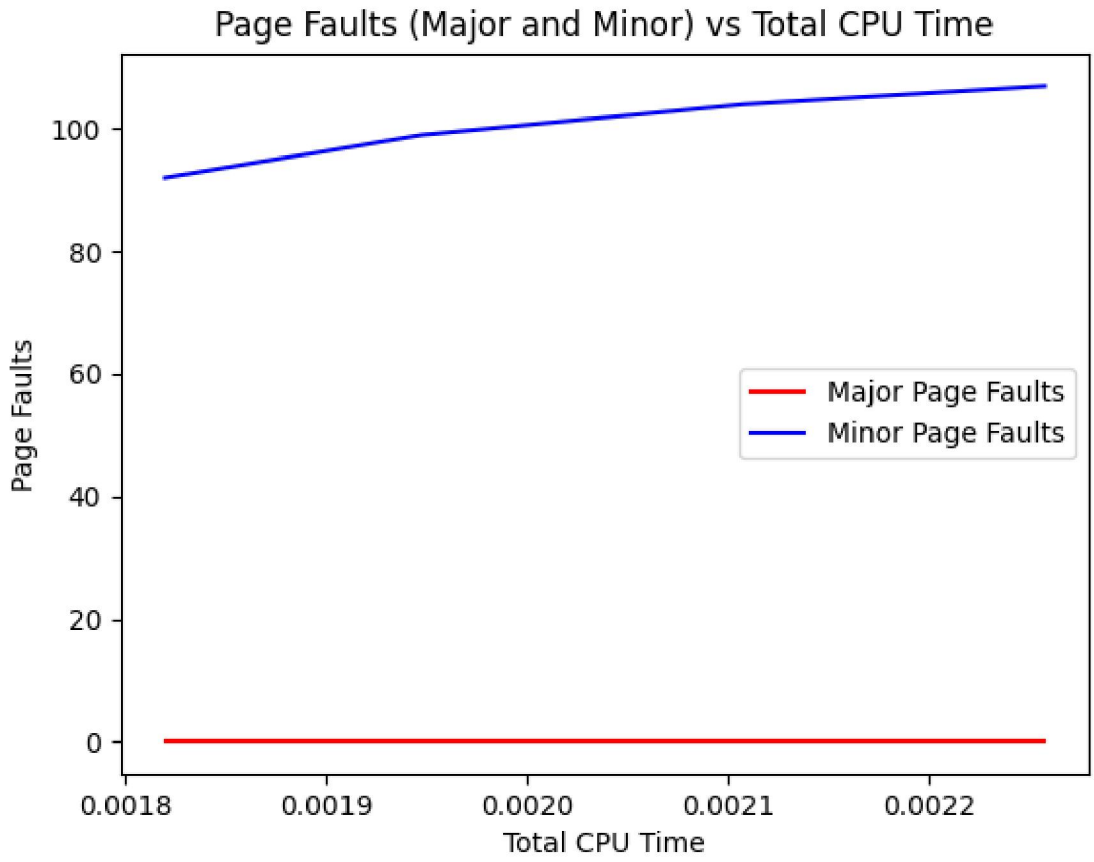
Command Line lab tool

1.

```
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4/build$ ./lab
----
beginning of main()
----
rusage:
User time: 0.001820
      System time: 0.000000
 Max resident set size: 1532
 Max page faults: 0
 Min page faults: 92

----
after ui_init()
----
rusage:
User time: 0.001948
      System time: 0.000000
 Max resident set size: 1532
 Max page faults: 0
 Min page faults: 99

Press Enter to exit...
----------------------
after starting UI thread
----------------------
rusage information
User time: 0.002108
      System time: 0.000000
 Max resident set size: 1532
 Max page faults: 0
```

```
 Min page faults: 104



----
after UI thread complete
----
rusage:
User time: 0.002258
      System time: 0.000000
 Max resident set size: 1532
 Max page faults: 0
 Min page faults: 107
```

2.



Comparing the results from the previous invocation, we can see that the user time, system time, max resident set size, major page faults, and minor page faults have all increased in this case as well. This is expected since the program is now executing with the LD_PRELOAD=gui/libgui.so environment variable set, which affects the resource usage.

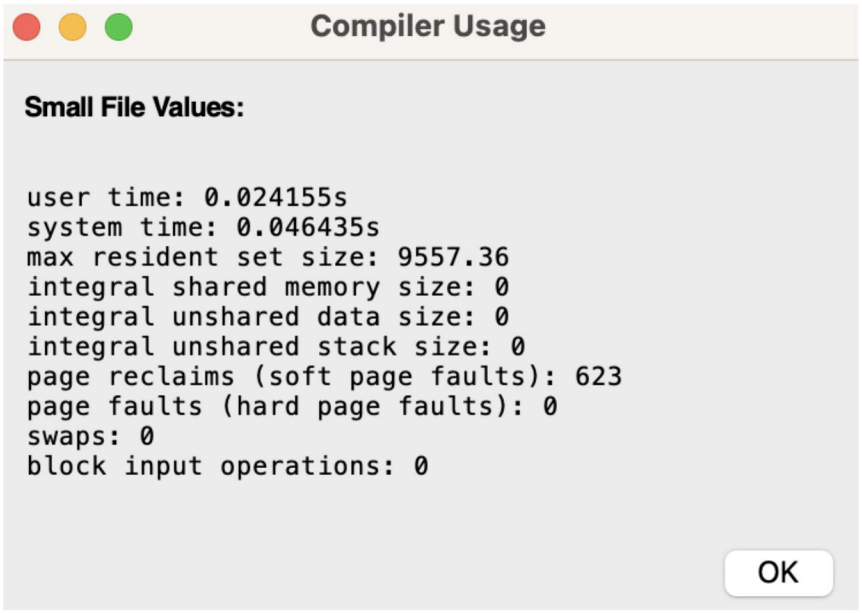
Hence, the implementation of get_command_rusage() effectively measures the resource usage of the given command by utilizing the wait4(2) function. The console output demonstrates an increase in resource usage when executing the program with LD_PRELOAD=gui/libgui.so, highlighting the impact of the environment variable on the program's performance.

3. Fixed on_click_run_grep_large and on_click_run_grep_small functions:

```
static void
on_click_run_grep_large(GtkWidget *_widget, gpointer _data)
{
    // Create a large file using dd command
    system("dd if=/dev/random of=large_file bs=1M count=100");

    char * const args[] = {
        "grep",
        "pattern", // Replace 'pattern' with the desired pattern to search
for
        "large_file",
        NULL
    };

    char buffer[4096];
    get_command_rusage(3, args, buffer, sizeof(buffer));

    dialog_show("Grep Large File Usage", "Executed grep on large_file",
buffer);
}

static void
on_click_run_grep_small(GtkWidget *widget, gpointer data)
{
    // Create a small file using dd command
    system("dd if=/dev/random of=small_file bs=1K count=100");

    char * const args[] = {
        "grep",
        "pattern", // Replace 'pattern' with the desired pattern to search
```

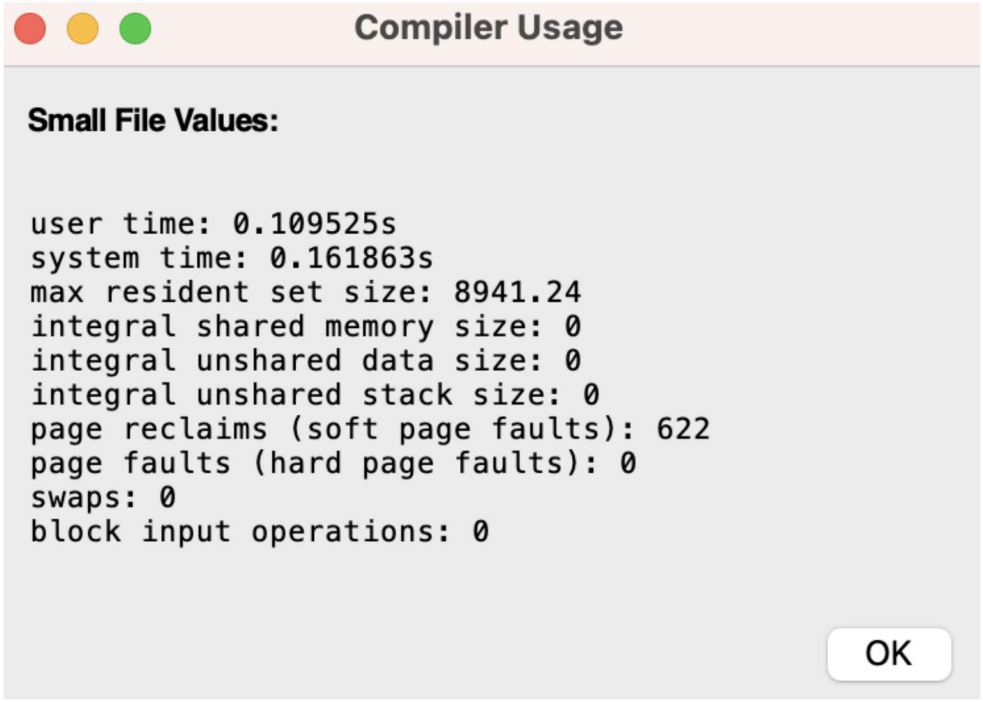
```
for
        "small_file",
        NULL
    };

    char buffer[4096];
    get_command_rusage(3, args, buffer, sizeof(buffer));

    dialog_show("Grep Small File Usage", "Executed grep on small_file",
buffer);
}
```

Values for small files:

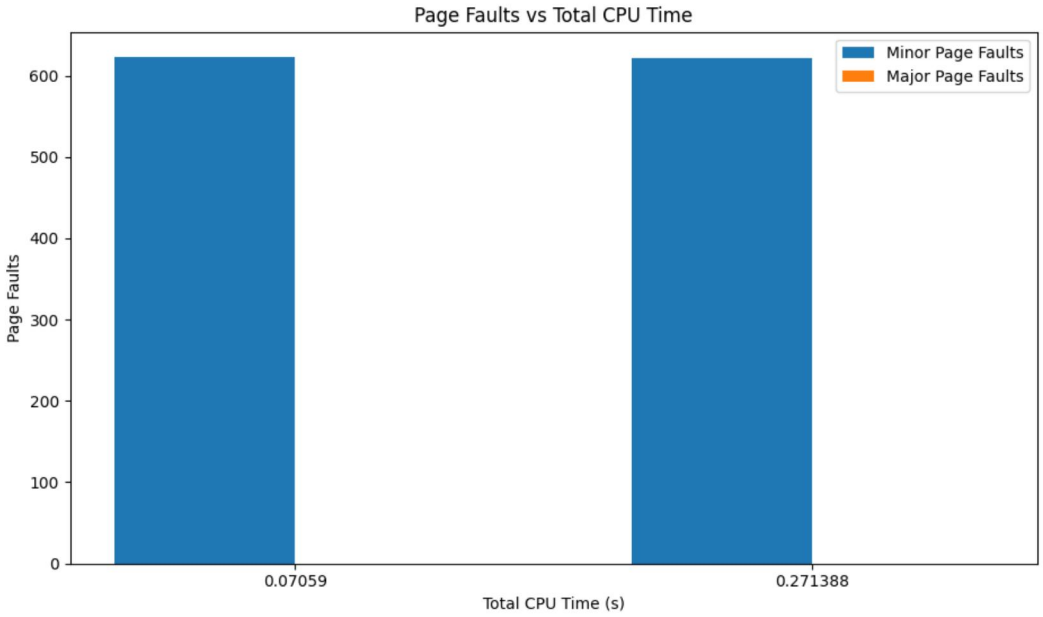



```
user time: 0.024155s
system time: 0.046435s
max resident set size: 9557.36
integral shared memory size: 0
integral unshared data size: 0
integral unshared stack size: 0
page reclaims (soft page faults): 623
page faults (hard page faults): 0
swaps: 0
block input operations: 0
block output operations: 0
```

Values for large files:




```
user time: 0.109525s
system time: 0.161863s
max resident set size: 8941.24
integral shared memory size: 0
integral unshared data size: 0
integral unshared stack size: 0
page reclaims (soft page faults): 622
page faults (hard page faults): 0
swaps: 0
block input operations: 0
block output operations: 0
```



Observations and differences:

- User time: The user time is higher for the large file compared to the small file, as grep needs more time to process the larger file.
- System time: Similarly, the system time is higher for the large file, as it requires more file system interaction to read the content.
- Max resident set size: The max resident set size for the small file is slightly higher than the large file. This difference could be due to variations in memory management and allocation during the execution of grep on different sized files. However, the difference is not significant and does not indicate a strong correlation between file size and memory consumption for grep.
- Page reclaims (soft page faults): The values are quite similar, with only a difference of 1, which indicates that the memory allocation and deallocation patterns are not significantly affected by the file size.

Overall, the main differences between running grep against small and large files lie in the user and system times, as expected due to the increased time required to process and interact with larger files. The memory consumption and page faults are not significantly impacted by the file size in this case.

3.3 on_click_run_*() **4 / 4**

✓ **- 0 pts** *Correct*

- 4 pts Not submitted / Failed to complete the section

- 2 pts Missing the plots

- 1 pts Missing the explanation

- 0.5 pts Not enough evaluation

- 1 pts Incomplete explanation

- 1 pts Missing evidence of the work and outputs

- 4 pts Not answered (or pages not linked)

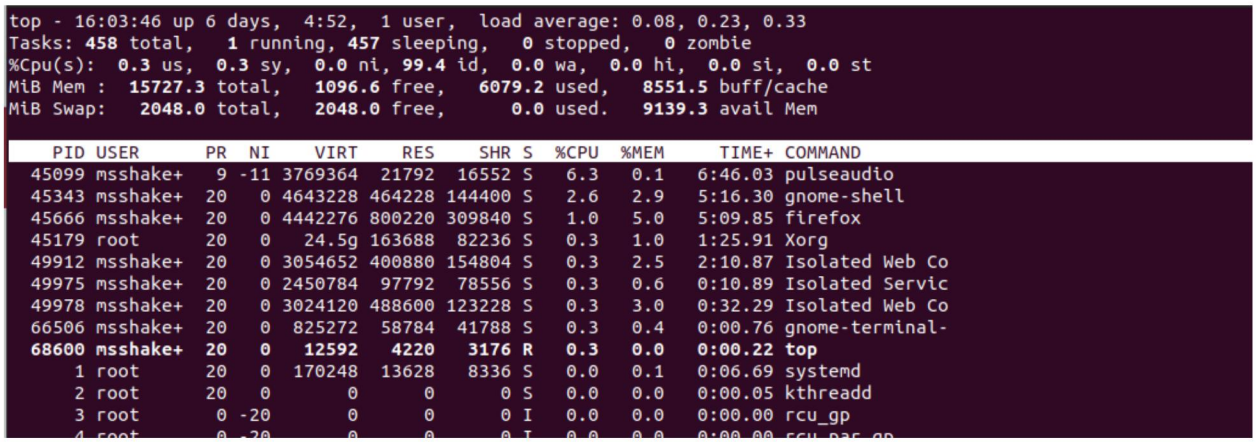
GUI

1.

```
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4/build$
LD_PRELOAD=gui/libgui.so ./lab
----
beginning of main()
----
rusage:
User time: 0.000000
      System time: 0.010450
 Max resident set size: 6368
 Max page faults: 0
 Min page faults: 479

----
after ui_init()
----
rusage:
User time: 0.007694
      System time: 0.023084
 Max resident set size: 15028
 Max page faults: 0
 Min page faults: 939

----
after UI thread complete
----
rusage:
User time: 0.051623
      System time: 0.035739
 Max resident set size: 24804
 Max page faults: 15
 Min page faults: 1659
```

Memory Pressure

1.



Current memory usage: 6079.2 MiB
Current memory free: 1096.6 MiB
Swap memory available: 2048 MiB

2.

```
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4/build$ /usr/bin/time -v gdb
./lab
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
<http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
      <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./lab...
(No debugging symbols found in ./lab)
(gdb) n
The program is not being run.
```

4.1 Memory availability **1 / 1**

✓ **- 0 pts** *Correct*

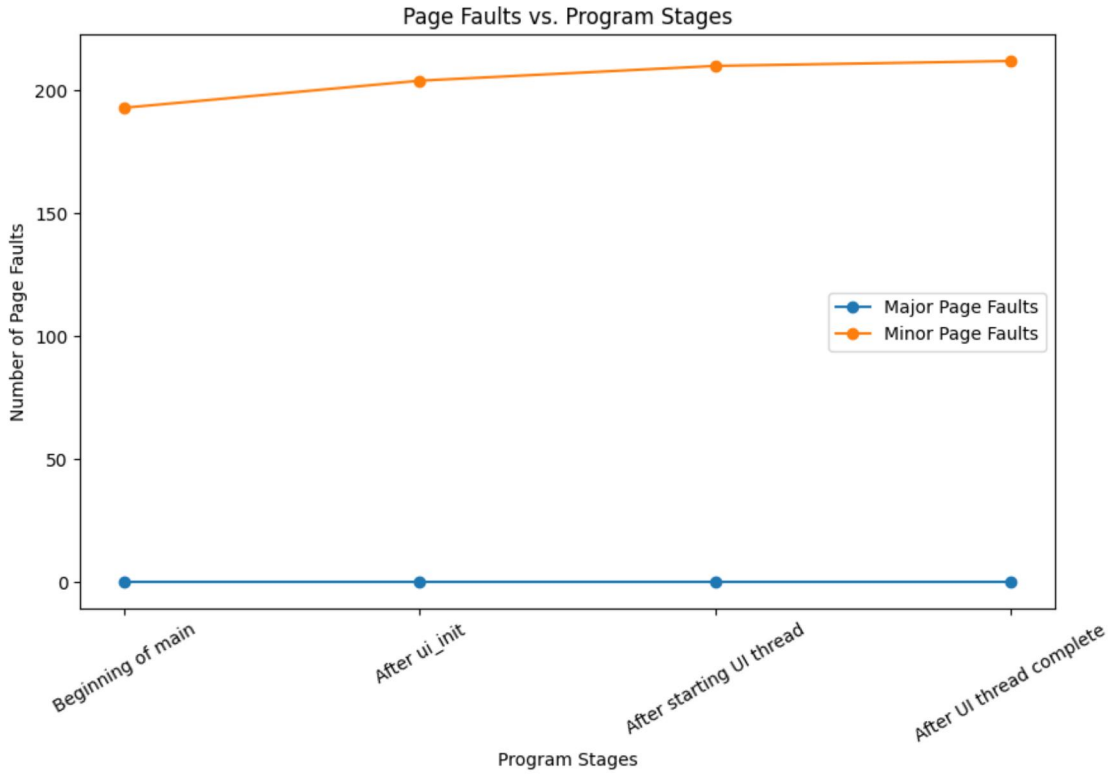
```
(gdb) run
Starting program: /users/labnet/st3/msshakeel/operatingLabs/lab4/build/lab
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
----
beginning of main()
----
rusage:
User time: 0.001623
      System time: 0.001623
 Max resident set size: 23416
 Max page faults: 0
 Min page faults: 193

----
after ui_init()
----
rusage:
User time: 0.001648
      System time: 0.001648
 Max resident set size: 23416
 Max page faults: 0
 Min page faults: 204

[New Thread 0x7ffff7d4f700 (LWP 69592)]
Press Enter to exit...
----------------------
after starting UI thread
----------------------
rusage information
User time: 0.001690
      System time: 0.001690
 Max resident set size: 23416
 Max page faults: 0
 Min page faults: 210



----
after UI thread complete
----
rusage:
User time: 0.001750
```

```
      System time: 0.001750
 Max resident set size: 23416
 Max page faults: 0
 Min page faults: 212

[Thread 0x7ffff7d4f700 (LWP 69592) exited]
[Inferior 1 (process 69586) exited normally]
(gdb) q
      Command being timed: "gdb ./lab"
      User time (seconds): 0.37
      System time (seconds): 0.06
      Percent of CPU this job got: 0%
      Elapsed (wall clock) time (h:mm:ss or m:ss): 3:03.45
      Average shared text size (kbytes): 0
      Average unshared data size (kbytes): 0
      Average stack size (kbytes): 0
      Average total size (kbytes): 0
      Maximum resident set size (kbytes): 51792
      Average resident set size (kbytes): 0
      Major (requiring I/O) page faults: 211
      Minor (reclaiming a frame) page faults: 14338
      Voluntary context switches: 424
      Involuntary context switches: 40
      Swaps: 0
      File system inputs: 42648
      File system outputs: 0
      Socket messages sent: 0
      Socket messages received: 0
      Signals delivered: 0
      Page size (bytes): 4096
      Exit status: 0
```

4.2 Run under gdb **2 / 2**

✓ **- 0 pts** *Correct*

- 1 pts Missing plots showing the page faults



From the graph it is observed that major page faults remain constant at 0 throughout the program's execution, while minor page faults increase from 193 at the beginning of main() to 212 after the UI thread is complete. This indicates that the program did not require any major page faults that involve disk I/O, but there were minor page faults that could be handled by reclaiming memory frames.

3. Output of Balloon.c:

```
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4$ ./ballonOut
Enter number of GiB to fill: 1
Memory usage before ballooning: 1484 KiB
ballonOut: mlock failed: Cannot allocate memory
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4$ ./ballonOut
Enter number of GiB to fill: 0.1
Memory usage before ballooning: 1484 KiB
Memory usage after ballooning: 1484 KiB
Ballooning complete. Press Enter to quit.
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4$ ./ballonOut
Enter number of GiB to fill: 2
```

```
Memory usage before ballooning: 1484 KiB
ballonOut: mlock failed: Cannot allocate memory
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4$ ./ballonOut
Enter number of GiB to fill: 0.9
Memory usage before ballooning: 1484 KiB
Memory usage after ballooning: 1484 KiB
Ballooning complete. Press Enter to quit.Q
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4$ ./ballonOut
Enter number of GiB to fill: 1.2
Memory usage before ballooning: 1484 KiB
ballonOut: mlock failed: Cannot allocate memory
```

There are a number of interesting observations here:

- When attempting to allocate 1 GiB of memory, the mlock function failed, indicating that the system could not allocate the requested memory.

- Allocating 0.1 GiB of memory did not result in any error, and the memory usage before and after ballooning remained the same at 1484 KiB. This is likely because the memory allocated was not large enough to cause a significant change in memory usage or the memory might not be locked successfully.

- Allocating 2 GiB of memory resulted in the same error as when trying to allocate 1 GiB, with mlock failing again.

- Allocating 0.9 GiB of memory did not result in any error, but memory usage before and after ballooning remained the same at 1484 KiB, similar to the case with 0.1 GiB.
- Allocating 1.2 GiB of memory resulted in the mlock function failing again.

These observations suggest that the system has a limit on the amount of memory that can be allocated, and the balloon application can't allocate beyond this limit. The mlock function fails when the system is unable to lock the requested memory, which occurs when attempting to allocate larger amounts of memory. I am fairly certain that the limit in this case is 1096.6 MiB.

4.3 Balloon out memory **1 / 1**

✓ **- 0 pts** *Correct*

- 1 pts Incomplete section/Failed to finish the section

- 1 pts Not answered (or pages not linked)

4.

```
msshakeel@slbncsf2112pc03:~/operatingLabs/lab4/build$ /usr/bin/time -v gdb
./lab
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
<http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
      <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./lab...
(No debugging symbols found in ./lab)
(gdb) run
Starting program: /users/labnet/st3/msshakeel/operatingLabs/lab4/build/lab
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
----
beginning of main()
----
rusage:
User time: 0.000000
      System time: 0.002655
 Max resident set size: 22832
 Max page faults: 0
 Min page faults: 193

----
after ui_init()
----
rusage:
User time: 0.000000
      System time: 0.002711
 Max resident set size: 22832
 Max page faults: 0
```

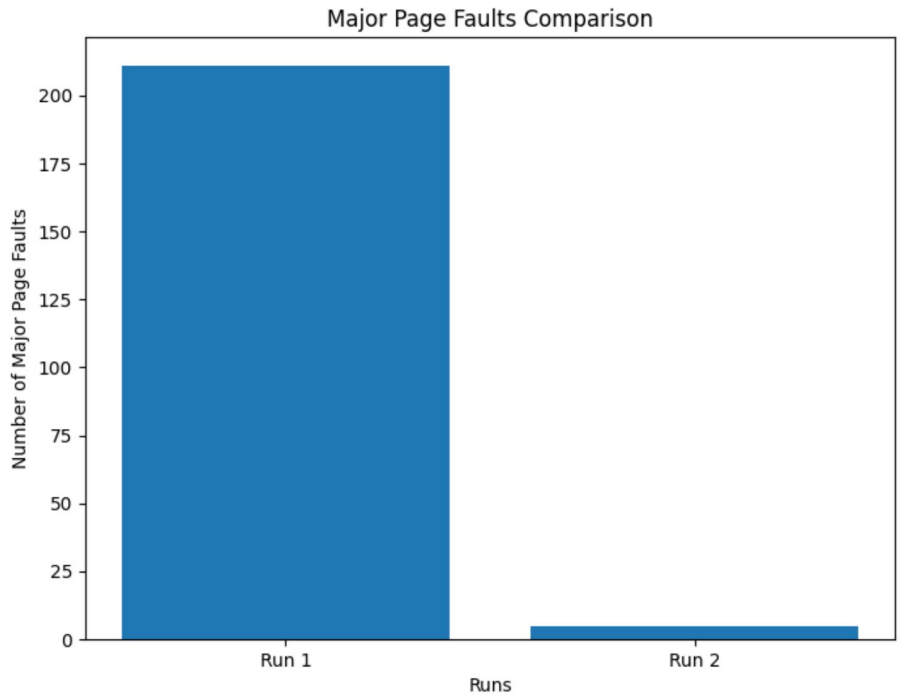
```
 Min page faults: 204

[New Thread 0x7ffff7d4f700 (LWP 71858)]
Press Enter to exit...
---------------------
after starting UI thread
---------------------
rusage information
User time: 0.000000
      System time: 0.002824
 Max resident set size: 22832
 Max page faults: 0
 Min page faults: 210



----
after UI thread complete
----
rusage:
User time: 0.000000
      System time: 0.002952
 Max resident set size: 22832
 Max page faults: 0
 Min page faults: 212

[Thread 0x7ffff7d4f700 (LWP 71858) exited]
[Inferior 1 (process 71853) exited normally]

(gdb) q
      Command being timed: "gdb ./lab"
      User time (seconds): 0.14
      System time (seconds): 0.01
      Percent of CPU this job got: 0%
      Elapsed (wall clock) time (h:mm:ss or m:ss): 2:26.10
      Average shared text size (kbytes): 0
      Average unshared data size (kbytes): 0
      Average stack size (kbytes): 0
      Average total size (kbytes): 0
      Maximum resident set size (kbytes): 50864
      Average resident set size (kbytes): 0
      Major (requiring I/O) page faults: 5
      Minor (reclaiming a frame) page faults: 11443
```

```
    Voluntary context switches: 68
    Involuntary context switches: 8
    Swaps: 0
    File system inputs: 288
    File system outputs: 0
    Socket messages sent: 0
    Socket messages received: 0
    Signals delivered: 0
    Page size (bytes): 4096
    Exit status: 0
```

In this run of the lab binary under gdb and /usr/bin/time -v, following differences in the page fault behavior compared to the previous run can be observed:

- Major (requiring I/O) page faults: In the current run, there are 5 major page faults, while in the previous run, there were 211 major page faults. This indicates that fewer page faults required disk I/O in the current run compared to the previous run.

- Minor (reclaiming a frame) page faults: In the current run, there are 11,443 minor page faults, while in the previous run, there were 14,338 minor page faults. This suggests that the current run required fewer page faults for reclaiming a frame compared to the previous run.



4.4 Run under gdb again **1 / 1**

✓ **- 0 pts** *Correct*

- 0.5 pts Missing the clarity in showing the major and minor page faults in the plot

- 0.5 pts Missing the representation of page faults

- 1 pts Missing the evidence of the work and outputs

- 1 pts Not answered (or pages not linked)