

Lab 10

Jarred Parr and Alexander Fountain

1. The total amount of available memory is `4gb` and the memory in use is `1.6gb`.
2. After launching reddit, netflix, and the weather channel, the memory usage went up to about `2.3db`. As a result, it is clear that, while heavily optimized for windows, Edge still has to eat some of the system ram in order to run complex web data. It's far less than the memory consumed by google chrome, however. The memory demand it about `.7gb`.
3. a. The standby is like the page buffering to allow a process to restart as soon as possible instead of waiting to be swapped in.
b. The amount of free memory before opening edge is 2388 MB. After opening the first instance of edge it drops to 2322 MB free but then drops again to 2239 MB free. After this stays consistent another instance of edge is opened and the free memory drops to 2063 MB but then goes back up to 2077 MB free.
4. The reason for the memory usage not being double that of the first instance is because the two programs share frames. Since double the frames are not needed, neither is double the memory.
5. Total physical memory is `4gb` and that total virtual memory is `5.37gb`. This is because there is a larger amount of virtual memory as a result of disk space almost always being larger than onboard memory. As a result, there is more available harddrive space to be used for virtual memory.
6. This value is `1408mb` and is used as an extension of RAM for data in RAM that has not been used recently. This is used to speed up reading data from the hard disk. This is similar to the linux swap file or swap partition, which does a similar utility of swapping idle processes out of RAM temporarily. This value of `1408mb` corresponds to the virtual memory size of `~5.4gb` because this is calculated as the total RAM + the pagefile size.
7. After using `RamMap.exe` it was seen that the `explorer.exe` program is `1484K`.
8. When we type there are no page faults but as soon as we click the menu to view the formatting options it spikes to 401 page faults. This is because the menu operation wasn't brought into memory origionally. Once the system stabalized and we changed the font from regular to bold italic, the page faults jumped to 8. This is a smaller page fault value than the menu operation because the program only has to bring in the font data and write it to memory. This write option does not cause any page faults.

Source Code

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

char* getState(int state){
    if(state == 4096) { // 0x1000
        return "COMMIT";
    } else if (state == 65536) { // 0x10000
        return "FREE";
    }
}
```

```
    } else if (state == 8192) { // 0x2000
        return "RESERVE";
    } else {
        return "ERROR";
    }
}

int main(){
    struct _SYSTEM_INFO info;
    GetSystemInfo(&info);
    printf("Page Size: %d\n", info.dwPageSize);
    char* c = malloc(1000000);
    struct _MEMORY_BASIC_INFORMATION memInfo;
    SIZE_T length;
    VirtualQuery(c, &memInfo, length);
    char* state = getState(memInfo.State);
    printf("%s\n", state);
    free(c);
    VirtualQuery(c, &memInfo, length);
    state = getState(memInfo.State);
    printf("%s\n", state);
}
```