1. **Maximum number of semaphores per process (static)**
   1. OS - Mac
   2. System call used - sysconf(), parameter - \_SC\_SEM\_NSEMS\_MAX
   3. Name of include file - unix standard header (unistd)
   4. Output



1. **Maximum value of a counting semaphore (static)**
   1. OS - Mac
   2. System call used - sysconf(), parameter - \_SC\_SEM\_VALUE\_MAX
   3. Name of include file - unix standard header (unistd)
   4. Output



1. **Maximum value of a counting semaphore (empirical)**
   1. OS - Linux (Ubuntu 22)
   2. System call used - semop with the sembuf incrementing the sem\_op section by one
   3. Name of include file - sys/sem.h
   4. Output



1. **Maximum size of a shared memory segment (empirical)**
   1. OS - Mac
   2. Used shmget() and shmctl() to create and remove shared memory segments while incrementing the size. From the <shm.h> library.
   3. Name of include file - shared memory facility (sys/shm)
   4. Output



1. **Page size in bytes (dynamic)**
   1. OS - Mac
   2. Used sysconf() with \_SC\_PAGESIZE to dynamically get the page size.
   3. Name of include file - unix standard header (unistd)
   4. Output



1. **Physical pages in a system (dynamic)**
   1. OS - Mac
   2. Used sysconf() with \_SC\_PHYS\_PAGES to get the number of physical pages.
   3. Name of include file - unix standard header (unistd)
   4. Output



1. **Maximum number of processes per user (dynamic)**
   1. OS - Mac
   2. Used sysconf() with \_SC\_CHILD\_MAX to determine max process per user
   3. Name of include file - unix standard header (unistd)
   4. Output



1. **Maximum filesize in bytes (dynamic)**
   1. OS - Linux (Ubuntu 22)
   2. Used getrlimit(RLIMIT\_FSIZE, &rlim)
   3. Name of include file - sys/resource.h
   4. Output



1. **Maximum number of open files, hard limit (dynamic)**
   1. OS - Linux (Ubuntu 22)
   2. Used getrlimit(RLIMIT\_OFILE, &rlim) and used the rlim\_max part of the struct
   3. Name of include file - unix standard header (unistd)
   4. Output



1. **Maximum number of open files, soft limit (dynamic)**
   1. OS - Linux (Ubuntu 22)
   2. Used getrlimit(RLIMIT\_OFILE, &rlim) and used the rlim\_cur part of the struct
   3. Name of include file - unix standard header (unistd)
   4. Output



1. **Clock resolution in milliseconds (dynamic)**
   1. OS - Linux (Ubuntu 22)
   2. Used sysconf(\_SC\_CLK\_TCK) to get the number of ticks per second then divided that number by 1000 to get milliseconds
   3. Name of include file - unix standard header (unistd)
   4. Output

