Authors: Brandon Conn and Mark Baker
10/25/2018
Performed on EOS15

System Object	Method	Value	Details
Maximum # of semaphores per process	static	256	Found in **posix1_lim.h** #define _POSIX_SEM_NSEMS_MAX 256
Maximum value of a (counting) semaphore	static	32767	Found in **posix1_lim.h** #define _POSIX_SEM_VALUE_MAX 32767
Maximum value of a (counting) semaphore	empirical	4294967295	sem_init(&semaphore, 0, 4294967296); causes an overflow

Maximum size of a shared memory segment (bytes)	empirical	~24600000000
		or shmmax
		18446744073692774399
		
		Kept varying everytime I would change it around this value
			int shmId;
			
			if ((shmId = shmget (IPC_PRIVATE, 24600000000,
			IPC_CREAT|S_IRUSR|S_IWUSR)) < 0) {
			perror ("i can't get no..\n");
			exit (1);
			}

Page size (bytes)	dynamic	4096	long page_size = 0;
			page_size = sysconf(_SC_PAGESIZE);
			printf("%ld\n", page_size);

Physical pages in system	dynamic	4078077	printf("This system has %ld pages of physical memory available.\n", get_phys_pages());
Maximum # of processes per user	dynamic	63663	printf(“%ld”, sysconf(_SC_CHILD_MAX));
Maximum filesize (bytes)	dynamic	-1	struct rlimit rl; getrlimit (RLIMIT_FSIZE, &rl);
Maximum # of open files: hard limit	dynamic	4096	struct rlimit rl; getrlimit (RLIMIT_NOFILE, &rl); printf("\n Hard Limit is : %lld\n", (long long int)rl.rlim_max);
Maximum # of open files: soft limit	dynamic	1024	struct rlimit rl; getrlimit (RLIMIT_NOFILE, &rl); printf("\n Soft Limit is : %lld\n", (long long int)rl.rlim_cur);

Clock resolution (msec)	dynamic	0.000001 msec	struct timespec res;
			
			rc = clock_getres(CLOCK_REALTIME, &res);
			
			printf("Clock Resolution: %ldns\n", res.tv_nsec);