

데이터사이언스를 위한 컴퓨팅2
HW1 report - 김정수 (2022-28981)

1.

1.1

(a)

stdio.h, math.h 파일 위치:
/usr/include

stdio.h의 경우 875줄 (마지막 enter 포함)이며,
math.h의 경우 math.h 자체는 1341줄 (마지막 enter 포함)이다.

(b)

```
494 extern int printf (const char *__restrict __format, ...);
495
542 extern int scanf (const char *__restrict __format, ...);
940 extern double sqrt (double __x) __attribute__((__nothrow__, __leaf__)); extern double _sqrt (double __x) __attribute__((__nothrow__, __leaf__));
```

(c)

포함되어 있지 않다.

Preprocessing 단계에서는 h.file을 갖고 오는데, 예를 들어 .c 상의 #include <math.h> 코드가 사라지고 그 부분이 math.h를 직접 갖고 온 부분으로 바뀐다. (헤더파일 확장) 이때 header file은 function과 이름들의 집합이기 때문에 그 안에 실제 함수가 어떻게 기능하는지에 대한 코드는 없는 상태이다.
따라서, 전처리된 결과에 sqrt코드는 포함되어 있지 않다.

1.2

(a) gcc -c sqrt.c

(b) object file ; ELF 64-bit LSB shared **object**, x86-64, version 1 (SYSV), not stripped

터미널 명령어: file sqrt.o

>> sqrt.o: ELF 64-bit LSB shared **object**, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2,

BuildID[sha1]=682b1bdd78417ce3d31c44f40d19b37f043bf062, for GNU/Linux 3.2.0, not stripped

1.3

(a) link를 하기 위해 필요한 명령어가 결여되어 있다. 따라서 이를 수정해주기 위해서는 gcc sqrt.c -o sqrt -lm 으로 바꿔줘야 한다.

(b)

```
shpc033@login0:~/hw1$ ./sqrt 2
1.41421356
```

2. (코드참고)

3.

(a)

node의 status 확인

즉, 슬럼 노드와 파티션이, 어떤 상태인지 알려준다.

출력으로는 shpc22라는 partition의 availability가 (up)이고, 프로그램이 돌아가는 시간의 limit이 1분이며 node 4개를 해당 파티션에서 쓰고 있으며, 그 노드의 상태는 idle(돌고 있고), 컴퓨터 노드들의 이름이 c[11-14](11~14까지 4개)인걸 알 수 있다.

```
shpc033@login0:~/hw1$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
shpc22      up          1:00      4    idle c[11-14]
```

(b)

제출된 job을 스케줄에서 확인하는 명령어. 출력: 아래 화면에서는 제출하지 않았기에 테이블이 안 채워진 상태이다. (slurm scheduling queue에 있는 job이 없다!)

jobid 는 submit한 작업의 id인데 현재 잡이 없으니 없고

partition은 해당 작업이 제출된 partition의 이름

name은 해당 작업의 이름

user는 작업을 submit한 리눅스계정 이름

st는 현재작업의 상태이다.

time 은 작업소요시간, nodes 그 작업에 쓰이는 노드개수, nodelist는 그 노드들이다.

```
shpc033@login0:~$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
shpc033@login0:~$
```

(c)

```
shpc033@login0:~$ srun -p shpc22 -N 2 hostname
c12
c11
```

slurm partition이 shpc22이라 하고, 2개 노드를 이용해서 hostname의 프로그램 job을 작동해라

출력: 해당하는 노드 2개 c12, c11

(d)

```

shpc033@login0:~$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
Address sizes:          46 bits physical, 48 bits virtual
CPU(s):                 64
On-line CPU(s) list:    0-63
Thread(s) per core:     2
Core(s) per socket:     16
Socket(s):               2
NUMA node(s):           2
Vendor ID:              GenuineIntel
CPU family:              6
Model:                   85
Model name:              Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz
Stepping:                7
CPU MHz:                 800.093
CPU max MHz:             3200.0000
CPU min MHz:             800.0000
BogoMIPS:                4200.00
Virtualization:          VT-x
L1d cache:               1 MiB
L1i cache:               1 MiB
L2 cache:                32 MiB
L3 cache:                44 MiB
NUMA node0 CPU(s):       0-15,32-47
NUMA node1 CPU(s):       16-31,48-63
Vulnerability Itlb multihit: KVM: Mitigation: Split huge pages
Vulnerability L1tf:       Not affected
Vulnerability Mds:        Not affected
Vulnerability Meltdown:   Not affected
Vulnerability Spec store bypass: Mitigation; Speculative Store Bypass disabled via prctl and seccomp
Vulnerability Spectre v1:  Mitigation; usercopy/swapgs barriers and __user pointer sanitization
Vulnerability Spectre v2:  Mitigation; Enhanced IBRS, IBPB conditional, RSB filling
Vulnerability Srbds:       Not affected
Vulnerability Tsx async abort: Mitigation; TSX disabled
Flags:                    fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe s
yscall nx pdpe1gb rdtscp lm constant tsc art arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf p
ni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid dca sse4_1 sse4_2 xzapic movbe popc
nt tsc deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch cpuid_fault epb cat_l3 cdp_l3 invpcid_single in
tel_ppin ssbd mba ibrs ibpb stibp ibrs_enhanced tpr_shadow vmx_i flexpriority ept vpid ept_ad fsgsbase tsc_adjust bmi1 avx
2 smep bmi2 erms invpcid cqm mpx rdt_a avx512f avx512dq rdseed adx smap clflushopt clwb intel_pt avx512cd avx512bw avx512
vl xsaveopt xsavec xgetbv1 xsaves cqm_llc cqm_occup_llc cqm_mbm_total cqm_mbm_local dtherm ida arat pln pts pku ospke avx
512_vnni md_clear flush_l1d arch_capabilities

```

```

shpc033@login0:~$ srun -p shpc22 -N 1 lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
Address sizes:          43 bits physical, 48 bits virtual
CPU(s):                 128
On-line CPU(s) list:    0-127
Thread(s) per core:     2
Core(s) per socket:     32
Socket(s):               2
NUMA node(s):           2
Vendor ID:              AuthenticAMD
CPU family:              23
Model:                   49
Model name:              AMD EPYC 7502 32-Core Processor
Stepping:                0
Frequency boost:         enabled
CPU MHz:                 1624.496
CPU max MHz:             2500.0000
CPU min MHz:             1500.0000
BogoMIPS:                5000.09
Virtualization:          AMD-V
L1d cache:               2 MiB
L1i cache:               2 MiB
L2 cache:                32 MiB
L3 cache:                256 MiB
NUMA node0 CPU(s):       0-31,64-95
NUMA node1 CPU(s):       32-63,96-127
Vulnerability Itlb multihit: Not affected
Vulnerability L1tf:       Not affected
Vulnerability Mds:        Not affected
Vulnerability Meltdown:   Not affected
Vulnerability Spec store bypass: Mitigation; Speculative Store Bypass disabled via prctl and seccomp
Vulnerability Spectre v1:  Mitigation; usercopy/swapgs barriers and __user pointer sanitization
Vulnerability Spectre v2:  Mitigation; Full AMD retpoline, IBPB conditional, IBRS_FW, STIBP conditional, RSB filling
Vulnerability Srbds:       Not affected
Vulnerability Tsx async abort: Not affected
Flags:                    fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext fx
sr_opt pdpe1gb rdtscp lm constant tsc rep_good nopl nonstop_tsc cpuid extd_apicid aperfmperf pni pclmulqdq monitor ssse3 fma cx16 sse4_1 sse4_2 movbe popc
nt aes xsave avx f16c rdrand lahf_lm cmp_legacy svm extapic cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw ibs skinit wdt tce topoext perfctr_core pe
rfctr_nb bpeext perfctr_llc mwaitx cpb cat_l3 cdp_l3 hw_pstate sme ssbd mba sev ibrs ibpb stibp vmcall fsgsbase bmi1 avx2 smep bmi2 cqm rdt_a rdseed adx s
map clflushopt clwb sha_ni xsaveopt xsavec xgetbv1 xsaves cqm_llc cqm_occup_llc cqm_mbm_total cqm_mbm_local clzero irperf xsaveerptr wbnoinvd arat npt lbr
v svm_lock nrip_save tsc_scale vmcb_clean flushbyasid decodeassists pausefilter pfthreshold avic v_vmsave_vmload vgif umip rdpid overflow_recov succor smc

```

첫번째 명령어는 로그인노드의 **cpu**에 대한 정보를 보여주는 것이고 두번째명령어는 계산노드의 **cpu**정보를 보여주는 것이다. 두 노드는 서로 다르고, 즉, 서로 다른 컴퓨터의 **cpu**의 정보이니 내용이 다른 것이다.