

Hw01

2024-27759 인문대학 언어학과 장효형

1 Compilation Process

1.1 Preprocessing

- (a) /usr/include/math.h와 /usr/include/stdio.h에 있다. 라인 수는 각각 1342, 876줄이다.(#endif 이후 빈 줄 하나가 더 있음)
- (b)

```
extern int printf (const char *__restrict __format, ...);

# 5 "sqrt.c"
void fallback_print_usage() {
    printf("Usage: ./sqrt number\n");
    printf("Example: ./sqrt 2\n");
    exit(0);
}

void print_sqrt(double number) { printf("%.8lf\n", sqrt(number)); }

int main(int argc, char *argv[]) {
    if (argc != 2) {
        fallback_print_usage();
    }
    print_sqrt(atof(argv[1]));
    return 0;
}
```

```
extern int scanf (const char *__restrict __format, ...) ;
```

```
extern int scanf (const char *__restrict __format, ...) __asm__ ("\" __isoc99_scanf")
```

```
# 3 "sqrt.c" 2
# 1 "/usr/include/stdlib.h" 1 3 4
# 25 "/usr/include/stdlib.h" 3 4
# 1 "/usr/include/x86_64-linux-gnu/bits/libc-header-start.h" 1 3 4
# 26 "/usr/include/stdlib.h" 2 3 4
```

```
extern double sqrt (double __x) __attribute__ ((__nothrow__ , __leaf__)); extern double __
```

(sou

```
# 4 "sqrt.c" 2
```

```
# 5 "sqrt.c"
```

```
void fallback_print_usage() {  
    printf("Usage: ./sqrt number\n");  
    printf("Example: ./sqrt 2\n");  
    exit(0);  
}
```

```
void print_sqrt(double number) { printf("%.8lf\n", sqrt(number)); }
```

- (c) 포함되어 있지 않다. printf, scanf, sqrt 함수는 extern이라는 키워드로 선언되고 있는데, extern은 외부에서 정의된 값을 불러 오는 키워드이다. 즉 외부 어느 파일에 실제로 printf, scanf, sqrt가 정의되어 있고, sqrt.p는 해당 파일에서 정의된 세 함수를 가져오는 역할만 한다.

1.2 Compilation

(a) gcc -c sqrt.c

(b) object file contains bytecodes(machine codes) which a machine can read, as well as metadata.(source: [Object file - Wikipedia](#), [What is the difference between assembly code and bytecode? - Stack Overflow](#))

1.3 linking

(a) 이유: math.h library는 실행파일을 만들 때 링커를 통해서 링킹해야 한다. 링커는 Object file과 외부 library를 연결해서 하나의 실행파일을 만드는 역할을 한다.

명령어: gcc sqrt.c -o sqrt.o -lm

```
shpc125@e1ogin3:~/hw1$ gcc sqrt.c -o sqrt.o -lm
```

```
shpc125@e1ogin3:~/hw1$ ./sqrt.o
```

```
Usage: ./sqrt number
```

```
Example: ./sqrt 2
```

```
shpc125@e1ogin3:~/hw1$ ./sqrt.o 4
```

```
2.00000000
```

```
shpc125@e1ogin3:~/hw1$ ./sqrt.o 3
```

```
1.73205081
```

(b)

2 C programming

2.1 Shift

(a) 1111 1111 1111 1111 1111 1111 1111 0000

(b) 1111 1111 1111 1111 1111 1111 1111 1100

(c) 0011 1111 1111 1111 1111 1111 1111 1100

(d) 두 숫자 모두 두 번의 shift가 일어난다. Arithmetic shift는 MSB를 보존한다. 주어진 예시에서 MSB가 1이므로 첫 번째 shift에서 MSB를 보존하고 MSB의 1을 오른쪽으로 밀어 채운 값인 1111 1111 1111 1111 1111 1111 1111 1000이 나타나게 되고, 두 번째 shift에서 또 MSB를 유지한 채 오른쪽으로 밀어 (b) 값이 나타난다. / 반면 logical shift는 MSB의 부호값과 무관하게 0을 채운다. 첫 번째 shift에서 0111 1111 1111 1111 1111 1111 1111 1000이 나타나고, 두 번째 shift에서 또 0을 채워 (c)의 값이 나타나게 된다. / 왼쪽으로 밀 경우, MSB와 무관하게 모두 0이 채워지므로 arithmetic shift와 logical shift는 동일하다.

2.2 Convert: 파일 제출로 대체

3 클러스터 사용 연습

```
shpc125@login3:~/hw1$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
class1      up           2:00      1    mix a04
class1      up           2:00      3  alloc a[00-02]
class1      up           2:00      8   idle a[03,05-11]
```

(a)

설명: node 정보를 확인한다. Alloc은 전체 CPU가, Mix는 일부 CPU만 node에 allocated되어 있다는 것을 말하며, idle은 하나도 할당되지 않은 것을 의미한다.

```
shpc125@login3:~/hw1$ squeue
JOBID PARTITION   NAME     USER ST       TIME  NODES NODELIST(REASON)
824029  class1 hostname shpc084 R        0:01     2 a[05-06]
```

(b)

설명: 어떤 작업이 진행중인지를 알려준다.

```
shpc125@login3:~/hw1$ srun -N 2 hostname
srun: job 824030 queued and waiting for resources
srun: job 824030 has been allocated resources
a06
a05
```

(c)

설명: -N은 사용하고자 하는 노드 개수를 의미한다. Srun으로 hostname 커맨드를

실행시키는데, hostname 커맨드는 자신이 할당된 노드의 이름을 반환한다.

```
shpc125@e1ogin3:~/hw1$ 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):                 32
On-line CPU(s) list:    0-31
Thread(s) per core:     2
Core(s) per socket:     8
Socket(s):              2
NUMA node(s):           2
Vendor ID:              GenuineIntel
CPU family:             6
Model:                  79
Model name:             Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz
Stepping:               1
CPU MHz:                1200.660
CPU max MHz:            3000.0000
CPU min MHz:            1200.0000
BogoMIPS:               4199.96
Virtualization:         VT-x
L1d cache:              512 KiB
L1i cache:              512 KiB
L2 cache:               4 MiB
L3 cache:               40 MiB
NUMA node0 CPU(s):      0-7,16-23
NUMA node1 CPU(s):      8-15,24-31
Vulnerability Itlb multihit: KVM: Mitigation: Split huge pages
Vulnerability L1tf:        Mitigation; PTE Inversion; VMX conditional cache flushes, SMT vulnerable
Vulnerability Mds:         Mitigation; Clear CPU buffers; SMT vulnerable
Vulnerability Meltdown:    Mitigation; PTI
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 generic retpoline, IBPB conditional, IBRS_FW, STIBP conditional, RSB filling
Vulnerability Srbds:       Not affected
Vulnerability Tsx async abort: Mitigation; Clear CPU buffers; SMT vulnerable
Flags:                    fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dcache
                           ts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant tsc

(d) shpc125@e1ogin3:~/hw1$ srun -N 1 lscpu
srun: job 824032 queued and waiting for resources
srun: job 824032 has been allocated resources
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:                800.665
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
```

설명: lscpu는 cpu 정보를 출력한다. lscpu와 srun -N 1 lscpu의 값이 다른 것은 srun -N 1을 통해 node로 들어갔고, 해당 node가 가지고 있는 cpu info가 뜨기 때문이다.