

과제 #1 : Linux의 명령어 실행을 통한 운영체제 기본 개념 이해

○ 과제 목표

- Linux 셸에서 몇 가지 간단한 명령어를 실행하여 운영 체제의 기본 개념 이해
- 과제에서 지시한 명령어를 실행하고 출력을 캡처해서 보고서를 작성(실행이란 단어가 들어가면 무조건 캡처해서 이를 보고서에 넣어야 함. 너무 길어 화면을 캡처하기 힘든 경우 일부만 캡처하거나 일반 파일로 redirection 하여 캡처해야 함)
- 과제에서 제시한 모든 질문을 답변해야 하며, 이를 위해 사용한 명령어(도구) 상세하게 명시
- 제공되는 파일 : cpu.c, cpu-print.c, disk.c, disk1.c, foo.pdf, make-copies.sh, memory1.c 및 memory2.c.

○ 기본 지식

- 프로세스 모니터링 도구(명령어)
 - ✓ top, ps 등
- /proc file system
 - ✓ Linux 커널이 시스템 및 프로세스에 대한 정보를 제공하는 특수한 형태의 파일 시스템
 - \$man proc 이나 인터넷에서 다양한 설명 확인 필요
 - /proc/meminfo 파일 이해
 - /proc/cpuinfo 파일 이해

○ 과제 내용

- 0. Linux 설치
 - ✓ 배포판 : Ubuntu (18.04 이상) 또는 Fedora (5.3 이상) 권장
 - ✓ 상세한 설치 방법은 인터넷 자료 활용
 - Dual Booting (Windows 및 Linux 동시 설치)
 - ☞ ※ 참고 : <https://coding-factory.tistory.com/494>
 - 가상 머신 (Virtual Box pro 또는 Vmware 등을 이용)으로 Linux 설치
 - ☞ ※ 참고 : <https://hiseon.me/linux/ubuntu/ubuntu-virtualbox-install/>
- 1. /proc 파일 시스템의 이해
 - ✓ (1) more / proc / cpuinfo 명령을 실행하고 processor, cores 등의 용어를 설명하라.
 - ※ 참고 : lscpu를 사용
 - ✓ (2) 학생들의 시스템에 있는 프로세스의 개수는 몇 개인가?
 - ✓ (3) 각 프로세스의 frequency는 얼마인가?
 - ✓ (4) 학생들이 컴퓨터 시스템의 물리적 메모리 크기는 얼마인가?
 - ✓ (5) 위 물리적 메모리 중 free한 크기는 얼마인가?
 - ✓ (6) 시스템 부팅 후 fork()된 프로세스의 개수는 몇 개인가?
 - ✓ (7) 부팅 후 문맥 교환(context switch)이 이뤄진 횟수는 얼마인가?
- 2. 실행 중인 프로세스의 상태 모니터링
 - ✓ (1) 주어진 cpu.c를 컴파일하고 아래 명령어를 실행하고 top 명령어 실행하라.
 - \$ gcc cpu.c -o cpu \$./cpu

- ※ 참고 : 무한 루프 수행 중 다른 터미널을 열고 top 명령을 실행
- ✓ (2) cpu 명령을 실행하는 프로세스의 PID는 무엇인가?
- ✓ (3) 위 프로세스는 얼마나 많은 CPU와 메모리를 소비하는가?
- ✓ (4) 위 프로세스의 현재 상태는 무엇인가? 예를 들어 실행 중, 차단(block) 중, 좀비 등
- 3. 새로운 자식 프로세스를 생성하여 사용자 명령을 실행하는 방법 이해
- ✓ (1) cpu-print.c를 컴파일하고 셸에서 아래 명령 차례로 실행하라.
 - \$ gcc cpu-print.c -o cpu-print
 - \$./cpu-print
- ※ 참고 : 무한 루프 수행 중 다른 터미널을 열고 ps 명령을 실행
- ✓ (2) cpu-print 프로세스의 부모, 즉 셸 프로세스의 PID는 무엇인가? 이 pid로부터 5세대 이상의 조상 프로세스 (또는 init 프로세스)에 도달 할 때까지 거슬러 모든 조상의 PID는 무엇인가?
- ✓ (3) \$./cpu-print> /tmp/tmp.txt & 을 실행하고 새로 생성 된 프로세스의 proc 파일 시스템 정보를 캡처하고 이를 설명하라. file descriptor 0, 1 및 2 (표준 입력, 출력 및 에러)가 가리키는 위치에 주의. 이러한 내용을 토대로, 셸에서 I/O 리디렉션을 구현하는 방법을 간단하게 설명하라.
- ✓ (4) \$./cpu-print | grep hello & 을 실행하고 새로 생성된 프로세스는 무엇인가? 또한 표준 입력, 표준 출력, 표준 에러가 가리키는 위치에 주의. 이러한 내용을 토대로, 셸이 파이프()를 구현하는 방법에 대해 간단하게 설명하라.
- ✓ (5) 셸에 명령어를 입력하면 셸은 (1) Linux에 내장된 명령을 실행(실행 파일 호출, built-in executable programs in the Linux kernel)하거나. (2) 내장된 실행 파일이 없는 경우, 셸은 자신의 코드 내에서 해당 명령어를 구현함. 이 때 cd, ls, history, ps 명령어 중 이미 Linux에 내장되어 실행 파일로 존재하여 bash 셸에 의해 실행되는 것은 무엇이고, bash shell 코드 자체에 의해 구현되어 있는 명령어는 무엇인가?
- 4. 프로세스의 가상 메모리와 실제 메모리 비교
- ✓ (1) 주어진 두 프로그램 memory1.c 및 memory2.c를 컴파일하고 실행하라.
 - ※ 참고 : 두 프로그램 모두 메모리에 큰 배열을 할당함. 한 프로그램은 할당한 배열에 접근하나 다른 프로그램은 할당한 배열에 접근하지 않음. 두 프로그램 모두 종료하기 전에 일시 중지하여 메모리 사용량을 검사(ps 명령어) 할 수 있음.
- ✓ (2) 각 프로세스의 가상메모리 크기와 실제메모리 크기를 비교하고, 주어진 프로그램 소스 코드와 ps 명령어를 통해 확인한 내용을 자세하게 설명하라.
- 5. 프로세스의 가상 메모리와 실제 메모리 비교
- ✓ (1) 주어진 disk.c 및 disk1.c 프로그램을 컴파일하고 실행하라.
 - ※ 참고 : 주어진 프로그램은 디스크에서 여러 파일을 읽음. 먼저 다음과 같이 파일을 생성해야 함. disk-files 디렉토리를 만들고 그 디렉토리에 foo.pdf 파일을 넣음. 주어진 make-copies.sh를 사용하여 해당 폴더에 다른 파일 이름으로 동일한 파일을 5000개 복사. disk 및 disk1 프로그램은 생성된 파일을 읽음.
- ✓ (2) disk.c 및 disk1.c 의 실행 파일 프로그램이 실행되는 동안 디스크 사용량을 측정하고, 주어진 프로그램 소스 코드와 iostat 명령어를 통해 확인한 내용을 자세하게 설명하라.

○ 과제 제출 마감

- 2020년 9월 9일 (수) 23시 59분 59초까지 구글클래스로 제출 (강의계획서 확인)

○ 필수 구현 (설명)

- 위 과제 내용 1, 2, 3, 4, 5를 모두 풀고 총 80점 이상 받은 경우 1번 과제를 완료한 것으로 함

○ 배점 기준

- 1. 14점
- 2. 12점
- 3. 20점
- 4. 24점
- 5. 30점