

과제 #1 : xv6 Tour

○ 과제 목표

- xv6 설치 및 컴파일
- “Hello xv6 World” 출력하는 응용 프로그램을 위한 구현 “helloxv6” helloxv6.c
- 두 번째 인자로 주어진 파일의 마지막 라인부터 공백 라인 포함 첫 번째 인자로 주어진 라인 수만큼 역순으로 출력하는 () “htac” htac.c 응용 프로그램을 위한 구현

○ 기본 지식

- xv6
 - ✓ 미국 에서 멀티프로세서 및 시스템을 위해 개발한 교육용 운영체제 MIT x86 RISC-V
 - ✓ UNIX V6 ANSI C 를 기반으로 구현
 - ✓ 리눅스나 와 달리 은 단순하지만 운영체제의 중요 개념과 구성을 포함하고 있음 BSD xv6 UNIX
- Cross Compile 방법 학습
 - ✓ xv6 gcc . xv6 에는 텍스트 편집기 또는 컴파일러가 없음 따라서 본 과제에서는 자신의 리눅스 환경에서 프로그램 작성 및 컴파일 후 생성된 실행파일을 상에서 수행함 , xv6

○ 과제 내용

1. xv6 설치 및 컴파일

✓ xv6 다운로드

```
$gitclonehttps://github.com/mit-pdos/xv6-public
```

✓ QEMU 다운로드 및 설치

- xv6 x86 QEMU x86 (운영체제는 자신의 컴퓨터에서 하드웨어를 에뮬레이트 하는 에뮬레이터에서 실행됨 에뮬레이터 없이도 운용 가능하나 수정을 위해 에뮬레이터 사용을 권장 ,)

```
$apt-getinstallqemu-kvm
```

✓ xv6 컴파일 및 실행

```
$make  
$makeqemu
```

```
( 1 ).makeqemu 예시 실행결과
```

```
root@oslab: /home/oslab/xv6-public
root@oslab:/home/oslab/xv6-public# make qemu
qemu-system-i386 -serial mon:stdio -drive file=fs.img,index=1,media=disk,format=raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -M 512
xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 minodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$
```

- 1 -

(2).ls 예시 실행결과

```
root@oslab: /home/oslab/xv6-public
$ ls
.          1 1 512
..         1 1 512
README    2 2 2206
cat        2 3 16244
echo       2 4 15100
forktest  2 5 9404
grep       2 6 18460
init       2 7 15680
kill       2 8 15128
ln         2 9 14980
ls         2 10 17612
mkdir      2 11 15224
rm         2 12 15204
sh         2 13 27844
stressfs   2 14 16116
usertest   2 15 67220
wc         2 16 16980
zombie     2 17 14792
console    3 18 0
$
$
$
$
```

2. helloxv6 응용 프로그램 작성

- “Hello xv6 World”를 출력하는 응용 프로그램 작성 helloxv6.c
- Makefile helloxv6.c make 을 수정하여 파일도 시 컴파일 되도록 변경
- xv6 , helloxv6 () 컴파일 실행 후 응용 프로그램의 실행파일 명령어 실행

(3).helloxv6.c 예시 파일작성예시

```
#include "types.h"
#include "stat.h"
#include "user.h"

int main(int argc, char** argv)
{
    printf(1, "Helloxv6World\n");
    exit();
}
```

(4).Makefile 1 예시 수정예시

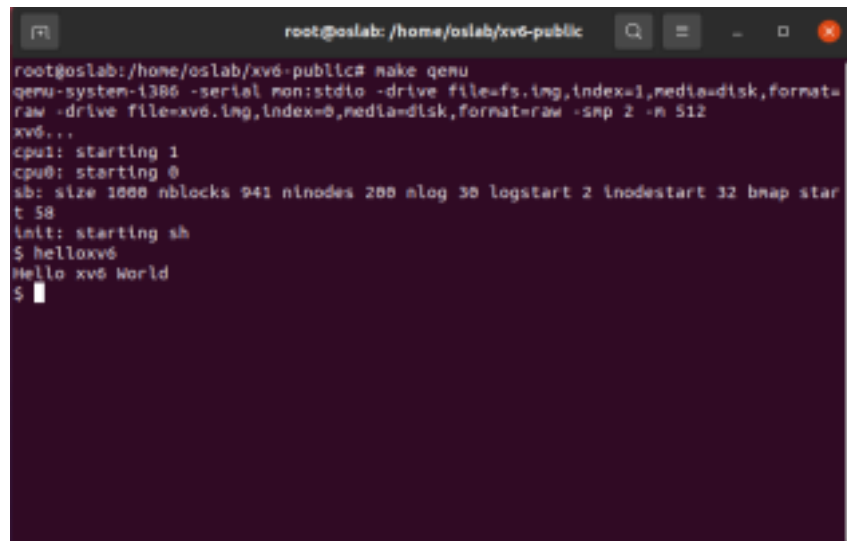
```
UPROGS=\
    _cat\
    _echo\
    _forktest\
    _grep\
    _init\
    _kill\
    _ln\
    _ls\
    _mkdir\
    _rm\
    _sh\
    _stressfs\
    _usertests\
    _wc\
    _zombie\
```

(5).Makefile 2 예시 수정예시

```
EXTRA=\
    mkfs.culib.cuser.hcat.cecho.cforktest.cgrep.ckill.c\
    ln.cls.cmkdir.crm.cstressfs.cusertests.cwc.czombie.c\
    printf.cumalloc.c\
    READMElist.txt\dot-bochsrc*.pltoc.*runoffrunoff1runoff.list\
    .gdbinit.tmplgdbutil\
```

- 2 -

(6).helloxv6 예시 실행결과



```
root@oslab: /home/oslab/xv6-public
root@oslab:/home/oslab/xv6-public# make qemu
qemu-system-i386 -serial mon:stdio -drive file=fs.img,index=1,media=disk,format=raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512
xv6...
cpu0: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 mlog 30 logstart 2 inodestart 32 bnep star
t 50
init: starting sh
$ helloxv6
Hello xv6 World
$
```

3. htac 쉘 프로그램

- 파일의 마지막 행부터 행의 내용을 역순으로 터미널에 출력하는 프로그램 작성 <n> htac.c - 기존의 파일을 참고하여 함수 구현 cat.c void htac(int fd);
- 단 은 에서 전역변수로 선언 , int line; main.c
- xv6 , htac () 컴파일 실행 후 응용 프로그램의 실행파일 명령어 실행

(7).htac.cmain 예시 함수부분소스코드

```

int main(int argc, char* argv[])
{
    int fd, i;

    if (argc <= 1)
    {
        cat(0);
        exit();
    }

    for( )
    {

    }
    exit();
}

```

- 3 -

(8).htac 예시 실행결과

```

root@oslab: /home/oslab/xv6-public
root@oslab: /home/oslab/xv6-public# make qemu
qemu-system-i386 -serial non:stdio -drive file=fs.img,index=1,media=disk,format=
raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -M 512
xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap star
t 58
init: starting sh
$ htac 5 README
simulator and run "make qemu".
Then run "make TOOLPREFIX=i386-jos-elf-". Now install the QEMU PC
x86 ELF binaries (see https://pdos.csail.mit.edu/6.828/).
will need to install a cross-compiler gcc suite capable of producing
"make". On non-x86 or non-ELF machines (like OS X, even on x86), you
$

```

(2022) Username: root, Password: 1234 참고 년 과제 부팅 시 를 입력하여 로그인 구현

✓ init.c

- 부팅 시 가장 먼저 실행하는 프로그램
- ssu_login을 호출하여 로그인 프로세스를 실행하도록 수정

(9).init.c (ssu_login) 예시 수정예시 구현필요

```

printf(1, "init: starting login\n");
pid = fork();
if (pid < 0) {
    printf(1, "init: fork failed\n");
    exit();
}
if (pid == 0) {
    exec("ssu_login", argv); // exec("sh", argv);
    printf(1, "init: exec login failed\n");
    exit();
}

```

✓ ssu_login.c

- Username Password (root, 1234) 과 를 입력으로 받음
- list.txt Username, Password 파일을 읽어 입력 받은 가 해당 파일에 존재하는지 확인 - Username Password , ssu_login shell (sh) fork-exec 과 가 존재한다면 에서 프로그램 을 으로 호출 - list.txt [Username] [Password] 파일은 형식으로 구성
- Makefile list.txt xv6 수정을 통해 파일을 에 추가

(10).list.txt (root1234/user3456 UsernamePassword) 예시 구성예시 은 로각한쌍

```
$catlist.txt
root1234
user3456
```

(11).ssu_login.c 예시 구성예시

```
#include"types.h"
#include"stat.h"
#include"user.h"
#include"fcntl.h"

charuserID[16][32];
charpwdID[16][32];

voidget_user_list(){
    intfd;

    fd=open("list.txt",O_RDONLY);
```

- 4 -

```
        for(i=0;i<10;i++){
            //list.txt Username,Password userID,pwdID 에서 정보를 에저장
        }
    }

    intcheck_idpw(){
    }

    intmain(intargc,char*argv[])
    {
        return0;
    }
```

(12). 예시 부팅시로그인실행결과



```
oslab@oslab: ~/xv6-public
oslab@oslab:~/xv6-public$ make qemu
qemu-system-i386 -serial mon:stdio -drive file=fs.img,index=1,media=disk,format=raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512
xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting login
Username: root
Password: 1234
$
```

○ 과제 제출 마감

- 2023 0 년 월 일 일 시 분 초까지 구글클래스 9 17 () 23 59 59 룸으로 제출
- 보고서 (hwp, doc, docx 3 () 등으로 작성 – 총 개의 프로그램이 수행된 결과 캡처 등이 포함된 와 소스코드 (helloxv6.c, htac.c, Makefile) . . (, xv6 를 제출해야 함 또한 별도로 학생이 구현한 프로그램 있으면 함께 제출 단 전체 소스코드는 제출 불필요.)
 - 마감시간 이후 시 24 . 0 . 30 간까지 지연 제출 가능 그 이후 제출은 점 처리 설계과제 마감시간 이후 지연 제출은 감 % . 점

○ 필수 구현 설치 및 설명 () 등

- 1, 2, 3

○ 배점 기준

- 1 : 30점
- 2 : 20점
- 3 : 50점