컴퓨터학부 20192403 박상철 / 비밀번호 : KARINA

1. 개요

교육용 OS인 xv6의 설치 및 사용법을 익히고 Makefile에 대해 분석해보는 시간을 가지며 helloxv6와 htac 구현을 통해 컴파일 방법과 xv6에서 지원하는 라이브러리 함수에 대해 배워가는 것을 목표로 함

2. 상세설계

2-1. make qeum 혹은 qemu-nox을 통해 xv6 운영체제에 접근

QEMU = qemu-system-i386

TOOLPREFIX = i686-elf-

위의 두 줄 주석 해제

UPROGS에 \_helloxv6, \_htac 추가

EXTRA에 helloxv6.c, htac.c 추가

2-2. helloxv6..c파일을 생성하여 컴파일

2-3. htac.c : 파일의 마지막 줄부터 올라가는 방향으로 한줄씩 출력

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

3. 결과

3-1텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

3-2

텍스트, 폰트, 화이트, 그래픽이(가) 표시된 사진

자동 생성된 설명

3-3 (과제 명세에 있는 테스트 예시)

텍스트, 스크린샷, 폰트, 블랙이(가) 표시된 사진

자동 생성된 설명

3-3(중간에 개행이 끼어있을 때도 동작하는지 테스트 예시 – README 잠시 수정)

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

4. 소스코드

helloxv6.c

|  |
| --- |
| #include "types.h"  #include "stat.h"  #include "user.h"  int main(int *argc*, char \*\**argv*)  {  printf(1, "Hello xv6 World\n");  exit();  } |

htac.c

|  |
| --- |
| #include "types.h"  #include "stat.h"  #include "user.h"  char \*buf;  int line;  //파일의 끝까지 읽어 파일의 전체 크기를 가져오는 함수  int  retFileSize(int *fd*)  {  int ret=0,n;  char tempBuf[512];  while((n = read(*fd*, tempBuf, sizeof(tempBuf))) > 0) {  ret+=n;  }  if(n<0) {  printf(1, "cat: read error\n");  exit();  }  return ret;  }  // 파일의 전체 크기를 받아 한번에 파일을 읽어들인 후 개행을 찾아 한 줄씩 출력  void  cat(int *fd*, int *fileSize*)  {  int n,cnt=0,end\_line;  buf = malloc(*fileSize*+1);  n = read(*fd*, buf, *fileSize*);  if(n != *fileSize*){  printf(1, "cat: read error\n");  exit();  }  end\_line = n;  for(int i=n-1; i >= 0; i--) {  if(buf[i] == '\n') {  write(1, &buf[i+1],end\_line-(i+1));  write(1, "\n", 1);  end\_line = i;  cnt++;  }  if(cnt == line) break;  }  free(buf);  }  int  main(int *argc*, char \**argv*[])  {  int fd, i, fileSize;  if(*argc* <= 2){  cat(0,0);  exit();  }  line = atoi(*argv*[1]);  if(line < 0) {  printf(1, "Lines can only be positive.\n");  exit();  }  for(i = *argc*-1; i >= 2; i--){  if((fd = open(*argv*[i], 0)) < 0){  printf(1, "cat: cannot open %s\n", *argv*[i]);  exit();  }  fileSize = retFileSize(fd);  close(fd);  if((fd = open(*argv*[i], 0)) < 0){  printf(1, "cat: cannot open %s\n", *argv*[i]);  exit();  }  cat(fd, fileSize);  close(fd);  }  exit();  } |