



دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

پروژه امتیازی ساختار و زبان کامپیوتر
مهندسی کامپیوتر

پروژه پنجم: کار با فایل x۳۹۰s

نگارش

پارسا نوروزی منش

آرش قوامی

هلنا دهخوارقانیان

امیرمحمد نصراله نژاد

استاد راهنما

استاد اسدی

بهمن ۱۴۰۳

فهرست مطالب

سوم	۱	کار با فایل ibm
سوم	۱-۱	مدیریت آدرس ها
سوم	۲-۱	دستور cd
سوم	۳-۱	دستور create
چهارم	۴-۱	دستور rename
چهارم	۵-۱	دستور mkdir
چهارم	۶-۱	دستور read
چهارم	۷-۱	دستور write
پنجم	۸-۱	دستور time
ششم	۲	چالش ها
ششم	۱-۲	مدیریت آدرس ها
ششم	۲-۲	دستور mkdir
ششم	۳-۲	دستور write

فهرست تصاویر

هفتم	برخی از ماکروها	۱-۲
هفتم	قسمت <code>data.</code>	۲-۲
هفتم	بخش <code>text.</code>	۳-۲

فصل ۱

کار با فایل ibm

۱-۱ مدیریت آدرس ها

در قسمت data. یک بافر خالی قرار دادیم. هنگامی که به هر فولدری وارد می شویم، در بافر آدرس اسم فولدر و سپس یک "/" به کمک تابع strcat زبان c اضافه می کنیم.

۲-۱ دستور cd

هنگام استفاده از این دستور با استفاده از دستور strcat اسم فولدر را به انتهای آدرس اضافه می کنیم و در آخر یک "/" اضافه می کنیم. دقت کنید که این کار به صورت خودکار دستور cd .. را هندل می کند. دقت کنید برای استفاده های دیگر مثلاً در دستور rename یک آدرس دوم را نیز نگه می داریم که مانند آدرس اول است.

۳-۱ دستور create

اسم فایلی که می خواهیم آن را بسازیم به انتهای آدرس اضافه می کنیم. (چون از قبل در انتهای آدرس "/" داشتیم دیگر لازم نیست دوباره آن را اضافه کنیم.) سپس آن را به ورودی های تابع fopen می دهیم و با صدا کردن این تابع فایل دلخواه ساخته می شود. دقت کنید همان طور که گفته شد برای احیای آدرس صحیح باید از انتها for بزنیم و تا وقتی که به اولین "/" برسیم همه کارکترها را با جایگزین می کنیم.

۴-۱ دستور rename

چون دو آدرس نگه داشتیم الان کافی است نام هر فایل را به یکی از هر دو آدرس بدهیم و سپس با صدا کردن تابع rename نام آنها را عوض کنیم. فقط دوباره مانند توابع قبل برای یافتن رشته اصلی در هر دو آدرس باید از انتها for بزنیم.

۵-۱ دستور mkdir

ابتدا در این تابع کد های ascii هر یک از حروف k و m و d و i و r ، را ه ترتیب در حافظه قرار می دهیم و سپس آدرس فعلی را در انتهای آن اضافه کرده و در نهایت نام فایل را به آن اضافه می کنیم. در ادامه با دادن این رشته به تابع system و صدا کردن آن دایرکتوری مورد نظر ساخته می شود.

۶-۱ دستور read

با پیدا کردن آدرس صحیح همانند دستورات قبلی فایل را باز می کردیم (به کمک تابع fopen و سپس به تعداد دفعات گفته شده از fgets استفاده کرده و خط ها را می خوانیم و چاپ می کنیم).

۷-۱ دستور write

اولین نکته این است که رشته ای که باید در فایل بنویسیم داخل ۲ کارکتر `newline` قرار دارد. پس ابتدا طول رشته را پیدا می کنیم و آخرین کارکتر را با کارکتر ۰ جایگزین می کنیم که کارکتر آخر از بین برود. سپس هنگامی که آدرس آنرا در %r ۳ قرار دادیم که به کمک تابع fprintf آن را در فایل قرار دهیم باید قبل از صدا کردن این تابع مقدار ثبات مورد نظر را با یک جمع کنیم که کارکتر اول هم از بین برود. نکته دیگر این است که باید مقدار رشته

n را با کارکتر newline جایگزین کند. برای اینکار از ابتدای رشته for می زنیم که به اولین اسلش برسیم که پس از این اسلش کارکتر n وجود دارد. سپس مقدار کارکتر ۰ را با اسلش جایگزین می کنیم و در نهایت به جای کارکتر n کارکتر ۰ قرار می دهیم. حالا کافی است ادامه رشته را برای اینکه از بین نرود با کمک strcat به انتهای آن اضافه می کنیم. در این صورت ۲ کارکتر اسلش و n را حذف کرده ایم و در نهایت یک کارکتر ۰ را به آن اضافه کردیم. در انتها صرفا کافی است این رشته را در فایل بنویسیم.

۸-۱ دستور `time`

برای این دستور صرفاً از چند تابع آماده C استفاده کردیم. از توابع `time` و `localtime` و `strftime` استفاده کردیم و در فرمت مناسب زمان را چاپ کردیم.

فصل ۲

چالش ها

۱-۲ مدیریت آدرس ها

چالش این قسمت این است که در تابع `strcat` وجود دارد این است که فقط دو آرگومان ورودی می گیرد و رشته دوم را با انتهای رشته اول اضافه می کند و این کار باعث می شود که رشته اول از بین برود. پس بعد از استفاده از این تابع باید یک `for` بنویسیم و از انتها همه ی کارکترهای اضافه شده را با کاراکتر `۰` جایگزین کنیم که دوباره به رشته اول برسیم.

۲-۲ دستور `mkdir`

علت اینکه کد اسکی هریک از حروف `mkdir` را در حافظه قرار دادیم و در همان ابتدا این مقدار را در حافظه قرار نداده ایم این بود که پس از اجرای اولین بار این دستور مقدار این رشته از بین می رفت!

۳-۲ دستور `write`

اولین مشکل در این قسمت این بود که ورودی رشته ای که قرار بود در فایل نوشته شود می توانست شامل کارکتر اسپیس باشد. برای انجام این کار به جای قرار دادن `%s` از فرمتی استفاده کردیم که `[%!]`!

شکل ۲-۱: برخی از ماکروها

شکل ۲-۲: قسمت data.

شکل ۲-۳: بخش text.