سپهر کیانیان

98102154 | نیم سال اول 99-98

پروژه مبانی برنامه سازی

مستند سازی پروژه

**توابع مرتبط با برنامه کلاینت :**

**1- تابع main() :**

در این برنامه تابع main() در واقع کد های مربوط به ظاهر برنامه و منو ها و چگونگی جابجایی از منویی به منوی دیگر را شامل می شود که تشخیص جابجایی از منویی به منوی دیگر نیز با تابع control() انجام می گیرد که در ادامه به بررسی آن می پردازیم.

ناگفته نماند که بیشتر توابعی که در برنامه client قرار داده شده اند و با سرور سر و کار دارند، با تشخیص اینکه پاسخ ارسال شده به سرور خطا است یا نه (اول error e دارد و با استفاده از آن برنامه متوجه می شود)، تصمیم می گیرند که در این منو بمانند یا به منوی دیگر بروند و تمام دستورات را با sprintf می سازند.

**2- تابع CSocket() :**

کار اصلی این تابع ساختن سوکت برای وصل شدن به سرور و ارسال پیام به سرور است و ورودی ندارد و خروجی آن client\_socket است که برای ارسال پیام به آن نیاز داریم و به توابع دیگر پاس داده می شود.

**3- تابع Reglog() :**

این تابع برای دو عمل register و login کردن ساخته شده است. (چون این دو شبیه به هم بودند، این دو را با یک تابع ساختم.) این تابع رشته اولیه دستور(که به ترتیب “register “ و “login “ است) را دریافت می کند و در تابع username و password را دریافت می کند و دستور مربوط را با sprintf می سازد و به سرور ارسال می کند.

**4- تابع CJChannel() :**

این تابع برای دو عمل create channel و join channel ساخته شده است. (باز چون این دو شبیه به هم بودند، این دو را با یک تابع ساختم.) این تابع authentication code مربوط به سرور و یک عدد برای این که تشخیص دهد کدام کار از دو کار بالا را انجام دهد، به عنوان ورودی دریافت می کند و با استفاده از sprintf دستور مربوط به هر کدام از این کار ها را می سازد و به سرور ارسال می کند.

**5- تابع Logout() :**

این تابع فقط authentication code به عنوان ورودی دریافت می کند و با استفاده از آن و sprintf دستور logout را می سازد و به سرور ارسال می کند.

**6- تابع SMessage() :**

این تابع نیز به عنوان ورودی فقط authentication code دریافت می کند و در تابع پیامی که کاربر می خواهد در کانال ارسال کند را دریافت می کند و دستور مربوطه را به سرور ارسال می کند.

**7- تابع RMLChannel() :**

این تابع برای سه کار refresh و channel members و leave ساخته شده است و به عنوان ورودی یک authentication code و یک عدد برای تشخیص اینکه کدام کار را انجام دهد، دریافت می کند و با توجه به عدد تصمیم میگیرد که کدام رشته را بسازد و سپس authentication code را به انتهای رشته می چسباند و به سرور دستور را ارسال میکند و اگر عمل leave نباشد، حتما پیام سرور را در output نشان می دهد.

**8- تابع Search() :**

این تابع برای دو عمل search user و search message ساخته شده است و فقط یک authentication code و یک عدد برای اینکه کدام کار را انجام دهد، به عنوان ورودی دریافت می کند و در تابع message یا user مربوطه را دریافت می کند(در تابع بین message و user تفاوتی وجود ندارد و هر دوی این دو را با یک دستور دریافت می کند) و سپس دستور مربوطه را با استفاده از authentication و user یا message می سازد.

**9- تابع checker() :**

این تابع برای ظاهر برنامه ساخته شده است و تنها کار آن این است که اگر دو آرگومان ورودی آن برابر بودند، کد اسکی مربوط به (<) و اگر برابر نبودند، کد اسکی مربوط به space را میفرستد.

**10- تابع control() :**

این تابع نیز برای ظاهر سازی برنامه ساخته شده است و در واقع برای تشخیص arrowkey ها و enter و درنتیجه برای کنترل روند برنامه با استفاده از این دو ساخته شده است و دو ورودی دریافت می کند و اگر کاربر arrowkey ای وارد کرد، با استفاده از این دو آرگومان، ظاهر برنامه تغییر می کند و اگر کاربر enter وارد کرد، عملیاتی انجام می دهد.

**11- تابع counter() :**

این تابع برای شمارش ساخته شده است. چون برای ظاهر سازی از کد clearscreen استفاده شده است، این تابع به اندازه مقدار زمانی که ورودی آن تعیین می کند، صفحه برنامه را freeze می کند و به کاربر زمان می دهد تا پیام مربوطه را ببیند.

**12- تابع backspacer() :**

این تابع نیز برای ظاهر برنامه و تابع کمکی counter است که به تعداد ارقام عدد ورودی اش backspace در خروجی وارد می‌کند.

**توابع مرتبط با سرور :**

**1- تابع main() :**

برنامه در تابع اصلی برنامه socket سرور را می سازد و socket کاربر را نیز دریافت می کند و دستور کاربر را دریافت می کند و سپس به توابع مختلف پاس داده می شود و کار مربوطه را انجام می دهد و سپس پاسخ ارسالی به سرور را نیز در output نشان می دهد. ناگفته نماند که تمام رشته های مربوطه (مخصوصا با قالب JSON) توسط sprintf و روش های ابداعی ساخته شده است و توسط sscanf آرگومان های قالب JSON دریافت شده است و تمام دستورات دریافتی ابتدا بررسی می شود که آیا درست ارسال شده اند یا نه و در دستورات مربوط به کاربر آنلاین، ابتدا بررسی می شود که کاربر آنلاین است یا نه.

**2- تابع reg() :**

این تابع ابتدا بررسی میکند که آیا username درخواستی کاربر قبلا در resources به وجود آمده و اگر به وجود آمده بود، خطا به کاربر ارسال می کند و اگر نبود، این username و password را در resources ایجاد می کند.

**3- تابع login() :**

این تابع در ابتدا بررسی می کند که آیا این username در resources موجود هست یا نه و سپس بررسی می کند که آیا کاربر مربوط به آن آنلاین هست یا نه و سپس بررسی می کند که آیا password مربوط به username معتبر هست یا نه و سپس کاربر را وارد linked list کاربران آنلاین می کند و با تابع authcreator برای آن کاربر authentication code غیر تکراری ارسال می کند.

**4- تابع CChannel () :**

این تابع در ابتدا بررسی می کند که آیا کانال مربوطه موجود هست یا نه و سپس فایل کانال را در resources می سازد و کانال را وارد linked list کانال های فعال می کند و کاربر را به کانال وصل می کند و کانال را نیز به کاربر وصل می کند و member های کانال را برابر یک قرار می دهد.

**5- تابع JChannel() :**

این تابع در ابتدا بررسی می کند که کانال در resources موجود هست و سپس پیدا می کند که کانال آنلاین هست یا نه و با استفاده از آن تصمیم میگیرد که چگونه کانال را به linked list کانال ها وصل کند و سپس کاربر را به کانال و کانال را به کاربر وصل میکند و member های مربوط به کانال را یکی افزایش می دهد.

**6- تابع Logout() :**

این تابع کاربر را از linked list کاربر های آنلاین حذف می کند و آن قسمت از حافظه که مربوط به این کاربر هست را free می کند.

**7- تابع SMessage() :**

این تابع بعد از تعیین کردن کانال داشتن یا نداشتن کاربر، پیام دریافتی از کاربر را با fprintf در فایل مربوط به کانال با فرمت JSON ارسال می کند.

**8- تابع Refresh() :**

در struct مربوط به کاربران آنلاین یک عدد filepos وجود دارد که نمایانگر این است که فرد پیام های مربوط به کانال را تا کجا خوانده است و تابع refresh بعد از پیدا کردن کاربر، پیام های از filepos به بعد را به کاربر ارسال می کند و filepos را برابر با مکان انتهای پیام های فایل - کانال قرار میدهد.

**9- تابع CMembers() :**

این تابع در ابتدا بررسی می کند که آیا کاربر عضو کانالی هست یا نه و با توجه به این که در هر کانالی یک linked list مربوط به member های آن موجود هست، به ترتیب name های member ها را یک رشته قرار می دهد و به کاربر ارسال می کند.

**10- تابع LChannel() :**

این تابع در ابتدا بررسی می کند که آیا کاربر عضو کانالی هست یا نه و سپس فرد را از linked list کانال ها حذف می کند و membersize کانال را یکی کم می کند و با توجه به این که کانال آیا کاربر عضو کانال دارد یا ندارد، تصمیم میگیرد کانال را از linked list کاربر های آنلاین حذف کند و حافظه مربوط به آن را free کند یا نه.

**11- تابع Searchuser() :**

این تابع در ابتدا بررسی می کند که آیا کاربر عضو کانالی هست یا نه و سپس بررسی می کند که آیا username مورد نظر با یکی از name های member های کانال مطابقت دارد یا نه و سپس با توجه به آن پیام موجودیت یا عدم موجودیت آن کاربر را به کاربر اعلام می کند.

**12- تابع Searchmessage() :**

این تابع در ابتدا بررسی می کند که آیا کاربر عضو کانالی هست یا نه و سپس فایل مربوط به کانال را باز می کند و یک به یک پیام های کانال را با تابع search بررسی می کند که آیا قسمتی از آن پیام این پیام را دارد(غیر از پیام های server به کانال) و سپس پیام هایی که در آن ها این پیام موجود بود را به کاربر ارسال می کند.

**13- تابع strcomp() :**

این تابع دو رشته به عنوان ورودی دریافت می کند و اگر این دو رشته هم از نظر طول رشته و character های رشته برابر بودند، 0 خروجی می دهد و در غیر این صورت، با توجه به این که طول کدام رشته بزرگتر است و یا در طول برابر اولین character نابرابر کدام بزرگتر است، خروجی 1 یا -1 می دهد.

**14- تابع authcreator() :**

این تابع با توجه به تعداد حداکثر اعضا تصمیم می گیرد که بررسی کند از ارقام آخر چند رقم تکراری نباید باشد و با توجه به آن تا آن قسمت را عدد تصادفی درست می کند و از آنجا به بعد، هر عدد تصادفی که درست می کند را بررسی می کند و اگر تکراری بود، دوباره عدد تصادفی درست می کند.

**15- تابع guserfinder() :**

این تابع در ابتدای اغلب توابع مربوط به کاربر آنلاین وجود دارد و در واقع کار اصلی آن این است که authentication code را از دستور کاربر دریافت می کند و بررسی می کند که آیا این authentication code معتبر است یا نه و با توجه به آن یک pointer به کاربر خروجی می دهد و یک کار دیگری که ممکن است در بعضی توابع انجام دهد، scan کردن یک رشته علاوه بر authentication code از دستور کاربر است.

**16- تابع gchannelchecker() :**

این تابع بررسی می کند که آیا کاربر عضو کانالی هست یا نه و اگر نبود، پیام خطای مربوط به آن را به کاربر ارسال می کند.

**17- تابع asscanf() :**

این تابع برای بعضی حالات خاص که sscanf نیز به کار نمی رود، به کار می رود و کار اصلی آن scan پیشرفته یک string از یک string دیگر است.

**18- تابع pointermover() :**

این تابع ۳ آرگومان به عنوان ورودی دریافت می کند که یکی pointer ای است که قرار است منتقل شود و یکی تعداد دفعاتی است که قرار است منتقل شود و آخری یک عدد است که تعیین می کند که این pointer، pointer به کانال است یا کاربر تا انتقال آن درست انجام شود.

**19- تابع search() :**

این تابع دو رشته ورودی دریافت می کند و اگر رشته دومی در اولی موجود بود، خروجی ۱ و اگر نبود، خروجی 0 می دهد.

**20- تابع out() :**

این تابع، تابع ظاهر برنامه است و برای نمایش دستورات دریافت شده و پاسخ های ارسال شده با یک قالب خاص است که همراه با ساعت و تاریخ دریافت یا ارسال دستور یا پاسخ را نمایش می دهد.