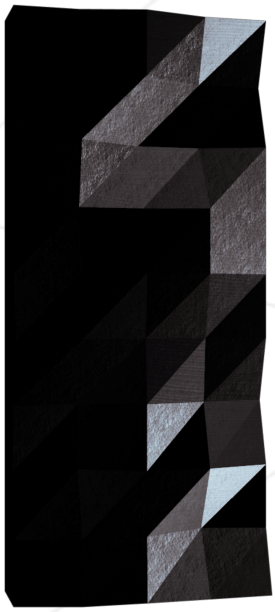

printf

Çünkü putnbr ve putstr yeterli değil

Özet: Bu proje gayet kolay. printf fonksiyonunu yeniden yazacaksınız. Bu fonksiyonu gelecek projelerde libft kütüphanenizin bir fonksiyonu olarak kullanabileceksiniz. Temelde değişken argümanları nasıl kullanacağınızı öğreneceksiniz.

Versiyon: 9

İçindekiler

Bölüm I

Giriş

`C` dilindeki `printf` fonksiyonunun çok yönlülüğü, bize programlama için çok büyük bir egzersiz sunuyor. Bu proje orta zorluktadır. Sizlerin `C` dilindeki değişken fonksiyonları keşfetmesini sağlayacaktır.
Başarılı `ft_printf` fonksiyonunun sırrı iyi yapılandırılmış ve kolay genişletilebilir koddur.

Bölüm II

Genel Talimatlar

- Projeleriniz C programlama dilinde yazılmalıdır.
- Projeleriniz Norm'a uygun olarak yazılmalıdır. Bonus dosyalarınız/fonksiyonlarınız varsa, bunlar norm kontrolüne dahil edilir ve bu dosyalarda norm hatası varsa `0` alırsınız.
- Tanımlanmamış davranışlar dışında sizin fonksiyonlarınız beklenmedik bir şekilde sonlanmamalıdır (Segmentasyon hatası, bus hatası, double free hatası, vb.) . Eğer bunlar yaşanırsa s `0` alırsınız.
- Heap'de ayırmış olduğunuz hafıza adresleri gerekli olduğu durumlarda serbest bırakılmalıdır. Hiçbir istisna tolere edilmeyecektir.
- Eğer verilen görev `Makefile` dosyasının yüklenmesini istiyorsa, sizin kaynak dosyalarınızı `-Wall`, `-Wextra` , `-Werror`, flaglarini kullanarak derleyip çıktı dosyalarını üretecek olan Makefile dosyasını oluşturmanız gerekmektedir. Makefile dosyasını oluştururken cc kullanın ve Makefile dosyanız yeniden ilişkilendirme yapmamalıdır (relink).
- `Makefile` dosyanız en azından `$(NAME)`, `all`, `clean`, `fclean` ve `re` kurallarını içermelidir.
- Projenize bonusu dahil etmek için Makefile dosyanıza `bonus` kuralını dahil etmeniz gerekmektedir. Bonus kuralının dahil edilmesi bu projenin ana kısmında kullanılması yasak olan bazı header dosyaları, kütüphaneler ve fonksiyonların eklenmesini sağlayacaktır. Eğer projede farklı bir tanımlama yapılmamışsa, bonus projeleri `_bonus.{c/h}` dosyaları içerisinde olmalıdır. Ana proje ve bonus proje değerlendirmeleri ayrı ayrı gerçekleştirilmektedir.
- Eğer projeniz kendi yazmış olduğunuz `libft` kütüphanesini kullanmanıza izin veriyorsa, bu kütüphane ve ilişkili `Makefile` dosyasını proje dizinindeki `libft` klasörüne ilişkili `Makefile` dosyası ile kopyalamanız gerekmektedir. Projenizin `Makefile` dosyası öncelikle `libft` kütüphanesini kütüphanenin `Makefile` dosyasını kullanarak derlemeli ardından projeyi derlemelidir.
- Test programları sisteme yüklenmek zorunda değildir ve puanlandırılmayacaktır. Buna rağmen test programları yazmanızı şiddetle önermekteyiz. Test programları

sayesinde kendinizin ve arkadaşlarınız projelerinin çıktılarını kolaylıkla gözlemleyebilirsiniz. Bu test dosyalarından özellikle savunma sürecinde çok faydalanacaksınız. Savunma sürecinde kendi projeleriniz ve arkadaşlarınızın projeleri için test programlarını kullanmakta özgürsünüz.

- Çalışmalarınız atanmış olan git repolarına yüklemeniz gerekmektedir. Sadece git reposu içerisindeki çalışmalar notlandırılacaktır. Eğer Deepthought sizin çalışmanızı değerlendirmek için atanmışsa, bu değerlendirmeyi arkadaşlarınızın sizin projenizi değerlendirmesinden sonra gerçekleştirecektir. Eğer Deepthought değerlendirme sürecinde herhangi bir hata ile karşılaşılırsa değerlendirme durdurulacaktır.

Bölüm III

Zorunlu kısım

Program adı	`libftprintf.a`
Teslim edilecek dosyalar	`*.c, */*.c, *.h, */*.h, Makefile`
Makefile	`all, clean, fclean, re,` bonus
Harici fonksiyonlar.	`malloc, free, write, va_start, va_arg, va_copy, va_end`
Libft kullanılabilir mi?	Evet
Açıklama	Gerçek printf fonksiyonunu taklit eden ft_printf fonksiyonunu içeren bir kütüphane yazın

- ft_printf fonksiyonunu prototipi `int ft_printf(const char *, ...);` şeklinde olmalıdır.
- `libc` kütüphanesinde bulunan `printf` fonksiyonunu yeniden yazmalısınız
- Yazacağınız fonksiyon gerçek `printf` gibi buffer yönetimi yapmamalıdır.
- Fonksiyonunuz şu dönüşümü yapmalıdır: `cspdiuxX%`
- Fonksiyonunuz gerçek printf ile karşılaştırılacaktır
- Kütüphanenizi oluşturmak için `ar` komutunu kullanmalısınız,`libtool` komutu kullanımı yasaklanmıştır.

Gerekli dönüşümler hakkında kısa açıklamalar:

- %c tek bir karakter yazdırır.
- %s bir karakter dizisi yazdırır.
- %p Void * pointer argümanını hexadecimal biçiminde yazdırır.
- %d 10 tabanında decimal sayı yazdırır.
- %i 10 tabanında tam sayı yazdırır.
- %u 10 tabanında işaretsiz decimal sayı yazdırır.
- %x hexadecimal sayıyı (16 tabanında) küçük harfler ile yazdırır.
- %X hexadecimal sayıyı (16 tabanında) büyük harfler ile yazdırır.
- %% yüzde işareti yazdırır.

`Daha fazlası için : man 3 printf / man 3 stdarg`

Bölüm IV

Bonus kısım

- Eğer zorunlu kısım harika değilse bonus için hayal kurmayın bile
- Tüm bonusları yapmak zorunda değilsiniz
- '`-0.`' flag'ini içeren tüm kombinasyonları ve tüm dönüşümler için minimum alan genişliğini yönetin
- '`# +`' flaglerini yönetin (evet bir tanesi boşluk)

`Eğer bonusları yapmayı planlıyorsanız, toy bir yaklaşımdan kaçınmalı ve bunları baştan nasıl yapacağınızı düşünmelisiniz.`