



printf

Çünkü ft_putnbr() ve ft_putstr() yeterli değil

Özet: Bu proje amacı gayet net. Printf() fonksiyonunu yeniden yazacaksınız. Genel olarak, değişen sayıda argüman nasıl kullanacağınızı öğreneceksiniz. Baya iyi güzel değil mi ?? Evet gerçekten güzel iyi :)

Versiyon: 10

İçindekiler

I	Giriş	2
II	Genel Talimatlar	3
III	Zorunlu kısım	5
IV	Bonus kısım	7
V	Proje Teslimi ve Akran Değerlendirmesi	8

Bölüm I

Giriş

Popüler ve çok kullanışlı bir C fonksyonu keşif edeceksiniz: `printf()`. Bu egzersiz, yazılım becerilerinizi geliştirmek için güzel bir fırsat. Bu proje orta zorluktadır.

Sizlerin C dilindeki **variadic fonksyonları** keşfetmesini sağlayacaktır.

Başarılı `ft_printf` fonksiyonunun sırrı iyi yapılandırılmış ve kolay genişletilebilir kodur.



Bu projeyi geçtikten sonra, kendi `ft_printf()` fonksiyonuzunu `libft` kütüphanenize ekleyip C dilinde okul projelerinde kullanabilirsiniz.

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 yaşanır 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**, flaglarını kullanarak derleyip çıktı dosyalarını üretecek olan **Makefile** dosyasını oluşturmanız gerekmektedir. **Makefile** dosyasını oluştururken **cc** kullanın ve **Makefile** dosyanız yeniden ilişkilendirme yapmamalıdır (re-link).
- **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	Makefile, *.h, /*.h, *.c, /*.c
Makefile	NAME, all, clean, fclean, re
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

libc kütüphanesinde bulunan `printf()` fonksiyonunu yeniden yazmalısınız. `ft_printf()` fonksiyonunu prototipi aşağıdaki şekilde olmalıdır:

```
int ft_printf(const char *, ...);
```

Proje için gereksinimler:

- Yazacağınız fonksiyon gerçek `printf` gibi buffer yönetimi yapmamalıdır.
- Fonksiyonunuz şu dönüşümleri yapmalıdır: `cspdiuxX%`
- Fonksiyonunuz gerçek `printf()` ile karşılaştırılacaktır.
- Kütüphanenizi oluşturmak için `ar` komutunu kullanmalısınız. `libtool` komutu kullanımı yasaklanmıştır.
- `libftprintf.a` repository'nizin kökünde oluşturulmalıdır.

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 integer 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

Bütün bonusları yapmak zorunda değilsiniz.

Bonus listesi:

-
- '-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.



Bonus kısım, yalnızca zorunlu kısım MÜKEMMEL ise değerlendirilecektir. MÜKEMMEL'den kastedilen, zorunlu olan kısım tamamı yapılmış ve sorunsuz çalışması demektir. TÜM zorunlu gereksinimleri geçmediyseniz, bonus bölümünüz hiç değerlendirilmeyecektir.

Proje Teslimi ve Akran Değerlendirmesi

Bu projeyi geçtikten sonra, kendi `ft_printf()` fonksiyonuzunu `libft` kütüphanenizine ekleyip C dilinde okul projelerinde kullanabilirsiniz.

